SEQUENTIAL ESTIMATION IN STATISTICS AND STEADY-STATE SIMULATION

A Thesis
Presented to
The Academic Faculty

by

Peng Tang

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and Systems Engineering

Georgia Institute of Technology
May 2014

SEQUENTIAL ESTIMATION IN STATISTICS AND STEADY-STATE SIMULATION

Approved by:

Professor Christos Alexopoulos,
Committee Chair
H. Milton Stewart School of Industrial
and Systems Engineering
Georgia Institute of Technology

Professor David Goldsman, Advisor
H. Milton Stewart School of Industrial
and Systems Engineering
Georgia Institute of Technology

Professor Jianjun Shi
H. Milton Stewart School of Industrial
and Systems Engineering
Georgia Institute of Technology

Professor Roshan Joseph Vengazhiyil
H. Milton Stewart School of Industrial
and Systems Engineering
Georgia Institute of Technology

Professor Xinwei Deng
Department of Statistics
Virginia Polytechnic Institute and State University

Date Approved: 3rd March 2014

To my family

ACKNOWLEDGEMENTS

First of all, I owe my deepest gratitude to my two advisors, Christos Alexopoulos and David Goldsman, who gave me invaluable guidance. I enjoyed every meeting and every conversation with them, and I will always remember their warm encouragement whenever I feel stressed. Also, their rigorous and pioneering attitudes towards research are perfect examples for me to learn.

I am also indebted to my co-authors, Xinwei Deng, James Wilson, and Huijing Jiang. I won't forget the days when we discussed and shared ideas together. They showed me how to find my own research problems, pursue the solutions, and validate the work. They also helped me a lot to improve my writing.

Friendship is another great harvest for me. I would like to thank all my friends at school and back in China. They are always there to share my laughs and tears. Thanks for the sincere wishes from them all.

Thanks to my parents Bingcheng Tang and Ti Luo, who unconditionally support and believe in me whenever and whatever. I owe a great gratitue to my boyfriend Lei Wang for his support and selfless love.

At last, I would like to thank the remaining members of my thesis committee, Jianjun Shi, Roshan Joseph Vengazhiyil, and Xinwei Deng, for their time and effort. Support for this work was provided by National Science Foundation grants CMMI-1233141 and CMMI-0927592.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

At the onset of the "Big Data" age, we are faced with ubiquitous data in various forms and with various characteristics, such as noise, high dimensionality, autocorrelation, and so on. The question of how to obtain accurate and computationally efficient estimates from such data is one that has stoked the interest of many researchers. This dissertation mainly concentrates on two general problem areas: inference for high-dimensional and noisy data, and estimation of the steady-state mean for univariate data generated by computer simulation experiments. We develop and evaluate three separate *sequential* algorithms for the two topics. One major advantage of sequential algorithms is that they allow for careful experimental adjustments as sampling proceeds. Unlike one-step sampling plans, sequential algorithms adapt to different situations arising from the ongoing sampling; this makes these procedures efficacious as problems become more complicated and more-delicate requirements need to be satisfied. We will elaborate on each research topic in the following discussion.

Concerning the first topic, our goal is to develop a robust graphical model for noisy data in a high-dimensional setting. Under a Gaussian distributional assumption, the estimation of undirected Gaussian graphs is equivalent to the estimation of inverse covariance matrices. Particular interest has focused upon estimating a sparse inverse covariance matrix to reveal insight on the data as suggested by the principle of parsimony. For estimation with high-dimensional data, the influence of anomalous observations becomes severe as the dimensionality increases. To address this problem, we propose a robust estimation procedure for the Gaussian graphical model based on the Integrated Squared Error (ISE) criterion. The robustness result is obtained by using ISE as a nonparametric criterion for seeking the

largest portion of the data that "matches" the model. Moreover, an l_1-type regularization is applied to encourage sparse estimation. To address the non-convexity of the objective function, we develop a sequential algorithm in the spirit of a majorization-minimization scheme. We summarize the results of Monte Carlo experiments supporting the conclusion that our estimator of the inverse covariance matrix converges weakly (i.e., in probability) to the latter matrix as the sample size grows. The performance of the proposed method is compared to that of several existing approaches through numerical simulations. We further demonstrate the strength of our method with applications in genetic network inference and financial portfolio optimization.

The second topic consists of two parts, both concerning the computation of point and confidence interval (CI) estimators for the mean μ of a stationary discrete-time univariate stochastic process $\{X_i : i = 1, 2, \ldots\}$ generated by a simulation experiment. The point estimation is relatively easy when the underlying system starts in steady state; but the traditional way of calculating CIs usually fails since the data encountered in simulation output are typically serially correlated. We propose two distinct sequential procedures that each yield a CI for μ with user-specified reliability and absolute or relative precision. The first sequential procedure focuses on variance estimators computed from standardized time series applied to nonoverlapping batches of observations, and it is characterized by its simplicity relative to methods based on batch means and its ability to deliver CIs for the variance parameter of the output process (i.e., the sum of covariances at all lags). The second procedure is the first sequential algorithm that relies solely on overlapping variance estimators to construct asymptotically valid CI estimators for the steady-state mean based on standardized time series. The advantage of the latter procedure is a substantial reduction in the average sample size required to deliver CIs under stringent precision requirements. The effectiveness of both procedures is evaluated via comparisons with state-of-the-art methods based on batch means under a series of experimental settings: the $M/M/1$ waiting-time process with 90% traffic intensity; the $M/H_2/1$ waiting-time process with 80% traffic intensity; the

$M/M/1/$*LIFO* waiting-time process with 80% traffic intensity; and an AR(1)-to-Pareto (ARTOP) process. We find that the new procedures perform comparatively well in terms of their average required sample sizes as well as the coverage and average half-length of their delivered CIs.

The thesis work is arranged into the following four chapters:

- **Chapter 2** proposes a robust nonparametric estimate for the Gaussian graphical model in high-dimensional situations.

- **Chapter 3** puts forward a new sequential algorithm based on standardized time series computed from nonoverlapping batches for computing point and CI estimators for the steady-state mean of a univariate simulation output process.

- **Chapter 4** designs and evaluates the second automated sequential procedure that uses overlapping variance estimators to construct asymptotically valid CI estimators for the steady-state mean of simulation output data.

- **Chapter 5** summarizes the main contributions and conclusions of this dissertation and discusses potential directions for future work.

CHAPTER II

ROBUST ESTIMATION IN SPARSE GAUSSIAN GRAPHICAL MODELS

2.1 Introduction

Gaussian graphical models have recently attracted a great deal of attention in various applications such as genetic networks and social networks. The Gaussian distribution is usually assumed as the underlying model because of its analytical convenience (Tibshirani 1996). Under this assumption, it is well known that the estimation of undirected Gaussian graphs is equivalent to the estimation of inverse covariance matrices (Lauritzen 1996).

Let $\mathbf{X} = (X^{(1)}, \ldots, X^{(p)}) \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ be a nondegenerate p-dimensional Gaussian random vector with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The inverse covariance matrix $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1} \equiv (c_{ij})_{1 \le i,j \le p}$ contains information related to the conditional dependency between variables. Each nonzero off-diagonal entry of $\boldsymbol{\Omega}$ represents an edge in the corresponding graph. If $c_{ij} = c_{ji} = 0$, then $X^{(i)}$ and $X^{(j)}$ are conditionally independent given the other variables $X^{(k)}$, $k \neq i, j$; otherwise, $X^{(i)}$ and $X^{(j)}$ are conditionally dependent. The problem originated from identifying significant genes that are responsible for certain diseases. There is an economic reality that gene studies are usually very costly; so it is our interest to be parsimonious in the number of genes to be selected, which leads to the focus on estimating a sparse inverse covariance matrix (Yuan and Lin 2007). The principle of parsimony also suggests that a sparse graph should be selected to adequately reveal insights about the data (Banerjee et al. 2006).

Various estimation methods for the covariance matrix have been proposed in the literature. Early developments concentrated on "shrinkage" approaches which shrink the eigenvalues of the sample covariance matrix to reduce the dispersion by minimizing the

expected quadratic loss (Haff 1977; Dey 1988; Perron 1992; Ledoit and Wolf 2003). However, those methods do not focus on the sparse structure of the inverse covariance matrix. For sparse inverse covariance matrix estimation, one direction is to exploit the connection between the entries of the inverse covariance matrix and the coefficients of the corresponding multivariate linear regression (Huang et al. 2006). Meinshausen and Bühlmann (2006) applied a neighborhood selection scheme by using lasso regressions of each variable against the remaining variables. Recently, likelihood-based methods have also attracted much attention. These methods identify zeros in the inverse covariance matrix by adding a sparsity-encouraged l_1 penalty to the objective based on the log-likelihood function. Numerous algorithms have been proposed to optimize the penalized log-likelihood function; see Banerjee et al. (2006), Yuan and Lin (2007), and Friedman et al. (2008), among others. Rather than directly using the likelihood function, Cai et al. (2011) investigated a constrained l_1 minimization method to estimate a sparse inverse covariance matrix as well as its convergence properties.

The above approaches can be effective when the observations are from the underlying multivariate normal distribution. If anomalous observations contaminate the data, then inference from a graphical model needs to be approached with caution. In the high-dimensional setting, where the dimension p is relatively large compared to the sample size n, it is not easy to check the underlying model assumptions. The existing likelihood-based methods may not be robust for inferring the graphical model since the likelihood function is sensitive to the presence of multivariate outliers. Since robustness is an important issue in the inference of high-dimensional graphical models, several methods have been proposed to address this concern. Peña and Prieto (2001) presented a multivariate outlier-detection procedure and a robust estimator for the covariance matrix based on the projection of the sample data to $2p$ selected dimensions. Miyamura and Kano (2006) introduced a robustified likelihood function to estimate the covariance matrix, and they also proposed corresponding test statistics associated with the robustified estimators. Finegold and Drton (2011)

proposed a tlasso method built upon Friedman et al.'s graphical lasso (glasso), but used the multivariate t-distribution for more-robust inferences of graphical models. Sun and Li (2012) developed an l_1 regularization procedure and corresponding coordinate descent algorithm to deal with robustness. Most of these methods are derived from the parametric perspective.

In this work, we investigate the robustness of Gaussian graphical models from a non-parametric perspective. Specifically, we adopt the Integrated Squared Error (ISE) criterion as a loss function in estimation, which seeks to find the largest portion of the data that "matches" the model (Scott 2001). Specifically, we aim to minimize the L_2 distance between the probability densities using the estimated and true inverse covariance matrices. This property naturally makes our proposed estimator more resistent to outliers compared with the likelihood-based approaches. Moreover, we encourage the sparsity of the estimator (have as few zeros as possible) by adding an l_1 penalty in the objective function. However, the resulting objective function is neither convex nor smooth. The traditional approach of relaxation into a constrained convex optimization appears to be computationally inefficient, especially in a high-dimensional space. To address this challenge, we develop an efficient algorithm from the majorization-minimization perspective, which is based on an upper bound for the original objective function. By minimizing the upper bound, the estimation problem can be transformed into an iterative estimation procedure in the context of graphical lasso (Friedman et al. 2008) with an appropriate weighting scheme on the observations. Moreover, the sequence of weights in each iteration is adaptively updated to automatically address potential anomalous observations.

The rest of the chapter is organized as follows. Section 2.2 reviews three representative methods in detail; in the later sections, we will compare our procedure with these methods. In Section 2.3, we introduce our robust inverse covariance matrix estimator and develop a computationally efficient algorithm to compute it. Section 2.4 provides a series of simulations to evaluate the performance of the proposed robust estimator. We continue

in Section 2.5 with two real-life applications using the proposed estimator. A discussion and concluding remarks are given in Section 2.6.

2.2 Literature Review

As we mentioned in the previous section, there are two major categories for estimating the inverse covariance matrix: shrinkage approaches and likelihood-based approaches. In particular, we elaborate on three representative methods, the LW method (Ledoit and Wolf 2003) which is a shrinkage approach, and two likelihood-based methods: the graphical lasso method (Friedman et al. 2008), and the tlasso method (Finegold and Drton 2011).

2.2.1 Ledoit-Wolf Method

Ledoit and Wolf (2003) proposed to estimate $\mathbf{\Sigma}$ by a linear combination of the sample covariance matrix $\mathbf{S}$ and the identity matrix $\mathbf{I}$,

$$\widehat{\mathbf{\Sigma}}_{LW} = (1-\nu)\mathbf{S} + \nu\tau\mathbf{I},$$

where the optimal values of τ and ν are obtained by minimizing the expected quadratic loss $E(\|\widehat{\mathbf{\Sigma}}_{LW} - \mathbf{\Sigma}\|_2)$. The corresponding estimate of $\mathbf{\Omega}$ is $\widehat{\mathbf{\Omega}}_{LW} = \widehat{\mathbf{\Sigma}}_{LW}^{-1}$.

2.2.2 Graphical Lasso Method

The graphical lasso (glasso) method was proposed by Friedman et al. (2008). This method estimates $\mathbf{\Sigma}^{-1}$ by maximizing

$$\log|\mathbf{\Sigma}^{-1}| - \mathrm{tr}[\mathbf{\Sigma}^{-1}\mathbf{S}] - \rho\sum_{i\neq j}|c_{ij}|,$$

where ρ is a positive tuning parameter.

The glasso method is an iterative algorithm that uses a block coordinate descent approach to solve the above optimization problem. To be specific, let $\boldsymbol{S}$ be the sample covariance matrix and $\boldsymbol{U}$ be an estimate of $\mathbf{\Sigma}$. Partition off the last row and column of $\boldsymbol{U}$

and $\boldsymbol{S}$ as

$$\boldsymbol{U} = \begin{pmatrix} \boldsymbol{U}_{11} & u_{12} \\ u_{12}^T & u_{22} \end{pmatrix}, \qquad \boldsymbol{S} = \begin{pmatrix} \boldsymbol{S}_{11} & s_{12} \\ s_{12}^T & s_{22} \end{pmatrix}.$$

By partially maximizing u_{12} with $\boldsymbol{U}_{11}$ fixed leads to $u_{12} = \boldsymbol{U}_{11}\boldsymbol{\beta}$, where $\boldsymbol{\beta}$ is the solution on $\mathbf{R}^p$ that minimizes

$$\frac{1}{2}\|\boldsymbol{U}_{11}^{1/2}\boldsymbol{\beta} - \boldsymbol{U}_{11}^{-1/2}s_{12}\|^2 + \rho\|\boldsymbol{\beta}\|_1.$$

The glasso method uses coordinate descent to obtain $\boldsymbol{\beta}$ in each of the coordinates $j = 1, \ldots, p-1$, with the update

$$\beta_j \leftarrow H\left(s_{jp} - \sum_{k \neq j} \boldsymbol{U}_{kj}\beta_k, \rho\right) / u_{jj},$$

where $H(\cdot)$ is the soft-threshold operator $H(x,t) = \text{sign}(x)(|x| - t)_+$. Then the algorithm cycles through the columns until convergence; see Friedman et al. (2008) for more details.

2.2.3 tlasso Method

Finegold and Drton (2011) proposed the tlasso method that estimates $\boldsymbol{\Sigma}^{-1}$ by modeling the data with a multivariate noncentral t-distribution $t_{p,\nu}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. They utilized the fact that if a multivariate normal random vector $\boldsymbol{X} \sim N_p(\mathbf{0}, \boldsymbol{\Sigma})$ is independent of a Gamma random variable $\tau \sim \Gamma(\nu/2, \nu/2)$, then $\boldsymbol{Y} = \boldsymbol{\mu} + \boldsymbol{X}/\sqrt{\tau}$ is distributed according to $t_{p,\nu}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Assume that the observations are $\{\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n\}$ and the corresponding hidden variables are $\{\tau_1, \ldots, \tau_n\}$; then Finegold and Drton formulated the penalized objective function as

$$\frac{n}{2}\log|\boldsymbol{\Sigma}^{-1}| - \frac{n}{2}\text{tr}\left(\boldsymbol{\Sigma}^{-1}S_{\tau YY}(\boldsymbol{\mu})\right) - \rho\left\|\boldsymbol{\Sigma}^{-1}\right\|_1,$$

with $S_{\tau YY}(\boldsymbol{\mu}) = \frac{1}{n}\sum_{i=1}^n \tau_i(\boldsymbol{Y}_i - \boldsymbol{\mu})(\boldsymbol{Y}_i - \boldsymbol{\mu})^T$. An EM algorithm (Dempster et al. 1977) is applied to iteratively maximize the above objective function. For the kth iteration, the E-step computes the value of $\{\tau_i^{k+1}\}$ using $\boldsymbol{\mu}^k$ and $(\boldsymbol{\Sigma}^{-1})^k$; and the M-step updates $\boldsymbol{\mu}^{k+1}$ and applies the glasso method to acquire $(\boldsymbol{\Sigma}^{-1})^{k+1}$.

2.3 Robust Estimation for Gaussian Graphical Model

In this section, we describe in detail the proposed robust estimator of the inverse covariance matrix. First, we will introduce the ISE criterion, which is the starting point for the proposed method.

2.3.1 Integrated Squared Error

ISE, also known as the L_2 minimizing criterion, seeks to find the minimal L_2 distance between the estimated probability density function (p.d.f.) and the true density. Its corresponding estimator is called L_2E. Rudemo (1982) initiated the criterion in the context of selecting the bandwidth of a histogram in kernel estimation. Scott (2001) investigated various parametric statistical models by using this criterion as a theoretical and practical estimation tool. In contrast to applying the local likelihood and the local least squares methods in nonparametric estimation (Fan and Gijbels 1996), Scott (2001) applied nonparametric techniques to the parametric models to study estimation robustness. In this work, we adopt the ISE criterion as a loss function for estimating the Gaussian graphical model.

Suppose $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are n observed p-dimensional random vectors, each of which follows a multivariate normal distribution $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with mean $\boldsymbol{\mu}$ and nonsingular covariance matrix $\boldsymbol{\Sigma}$. Without loss of generality, we assume that $\boldsymbol{\mu} = \mathbf{0}$. We consider a method based on minimizing the following L_2 distance as a function of $\boldsymbol{\Omega}$, the positive definite estimator of the true inverse covariance matrix $\boldsymbol{\Omega}^*$:

$$\begin{aligned}
&\int_{\mathbf{R}^p} (\phi(\mathbf{x}|\boldsymbol{\Omega}) - \phi(\mathbf{x}|\boldsymbol{\Omega}^*))^2 \, d\mathbf{x} \\
&= \int_{\mathbf{R}^p} \phi^2(\mathbf{x}|\boldsymbol{\Omega}) \, d\mathbf{x} - 2 \int_{\mathbf{R}^p} \phi(\mathbf{x}|\boldsymbol{\Omega}) \phi(\mathbf{x}|\boldsymbol{\Omega}^*) \, d\mathbf{x} + \int_{\mathbf{R}^p} \phi^2(\mathbf{x}|\boldsymbol{\Omega}^*) \, d\mathbf{x}, \qquad (2.3.1)
\end{aligned}$$

where $\phi(\mathbf{x}_i|\boldsymbol{\Omega})$ is the $N_p(\boldsymbol{\mu}, \boldsymbol{\Omega}^{-1})$ p.d.f. It is possible to find $\boldsymbol{\Omega}$ that minimizes an unbiased estimate of the distance. Note that the third term on the right-hand side of Equation (2.3.1) does not depend on $\boldsymbol{\Omega}$. Also, the second integral in Equation (2.3.1) can be viewed as $E(\phi(\mathbf{X}|\boldsymbol{\Omega}))$. With the observed data available, one can approximate $E(\phi(\mathbf{X}|\boldsymbol{\Omega}))$ by the

empirical mean $\frac{1}{n}\sum_{i=1}^{n}\phi(\mathbf{x}_i|\mathbf{\Omega})$. Such an approximation is commonly used in other related work using ISE (Scott 2001; Chi and Scott 2012). Now we reformulate Equation (2.3.1) to obtain the loss function as

$$\begin{aligned} L_n(\mathbf{\Omega}) &\equiv \int_{\mathbf{R}^p}\phi^2(\mathbf{x}|\mathbf{\Omega})\,d\mathbf{x} - 2\int_{\mathbf{R}^p}\phi(\mathbf{x}|\mathbf{\Omega})\phi(\mathbf{x}|\mathbf{\Omega}^*)\,d\mathbf{x} \\ &\approx \int_{\mathbf{R}^p}\phi^2(\mathbf{x}|\mathbf{\Omega})\,d\mathbf{x} - \frac{2}{n}\sum_{i=1}^{n}\phi(\mathbf{x}_i|\mathbf{\Omega}) \\ &\propto \frac{1}{2^{p/2}}|\mathbf{\Omega}|^{1/2} - \frac{2}{n}|\mathbf{\Omega}|^{1/2}\sum_{i=1}^{n}\exp\Big(-\frac{1}{2}\mathbf{x}_i^T\mathbf{\Omega}\mathbf{x}_i\Big), \end{aligned} \tag{2.3.2}$$

up to constant $(2\pi)^{-p/2}$.

Basu et al. (1998) described a family of divergences in which L_2E is a special case and MLE is a limiting one. They further pointed out that the MLE is the most-efficient estimator but lacks robustness. Compared to the MLE, L_2E illustrates the tradeoff between the efficiency and the robustness. Scott (2001, 2004) demonstrated that this tradeoff in asymptotic efficiency is similar to that seen in comparing the mean and median as a location estimator. Later, Scott (2011) showed that L_2E has the advantage of a simpler and faster computational solution compared to other members in the family.

To encourage sparsity for the purpose of inverse covariance matrix estimation, one straightforward strategy is to impose an l_1 regularization on the objective function, Equation (2.3.2). This is similar to the idea of the lasso in linear regression (Tibshirani 1996) and was also adopted by Yuan and Lin (2007) for Gaussian graphical model estimation. Specifically, we can obtain a positive definite estimator $\mathbf{\Omega}$ of the true inverse covariance matrix $\mathbf{\Omega}^*$ by minimizing

$$\begin{aligned} L_{n,\rho_n}(\mathbf{\Omega}) &\equiv L_n(\mathbf{\Omega}) + \rho_n\|\mathbf{\Omega}\|_1 \\ &\propto 2^{-p/2}|\mathbf{\Omega}|^{1/2} - \frac{2}{n}|\mathbf{\Omega}|^{1/2}\sum_{i=1}^{n}\exp\Big(-\frac{1}{2}\mathbf{x}_i^T\mathbf{\Omega}\mathbf{x}_i\Big) + \rho_n\|\mathbf{\Omega}\|_1, \end{aligned} \tag{2.3.3}$$

where $\|\mathbf{\Omega}\|_1 = \sum_{i,j}|c_{ij}|$ is an l_1 penalty. The purpose of the tuning parameter ρ_n is to balance the goodness of fit and sparsity of the model. It is a function of the sample size

n, and $\sqrt{n}\rho_n$ goes to some constant ρ_0 as $n \to \infty$. The motivation for using the objective function in Equation (2.3.3) is to provide robust estimates by utilizing the ISE criterion and to employ the l_1 penalty to encourage sparsity.

There is a computational challenge encountered in minimizing the objective function in Equation (2.3.3), due to its nonconvexity and nonsmoothness. To overcome these obstacles, one possible solution seeks to convert the problem into a constrained convex optimization problem. The key idea is to approximate the objective function as $e^{t/2} + \rho_n \|\mathbf{\Omega}\|_1$ subject to $\log|\mathbf{\Omega}| \leq t$, with additional constraints

$$1 - \frac{2^{1+p/2}}{n} \sum_{i=1}^{n} \exp(-\tfrac{1}{2}\mathbf{x}_i^T \mathbf{\Omega} \mathbf{x}_i) \;\geq\; \beta, \quad \text{where } \beta = -\frac{2}{(2\pi)^{p/2}} + \frac{1}{2^p \pi^{p/2}}.$$

The detailed derivation is found in Appendix A. However, such an implementation is computationally inefficient for solving a constrained optimization problem, especially when the dimensionality p gets large. To address this problem, we propose a novel method of approximating the objective function in Equation (2.3.3) from a majorization-minimization perspective. The main idea can be viewed as a general class of the EM algorithm, which operates by iteratively minimizing a surrogate function that majorizes the objective function (Hunter and Lange 2004). Moreover, such an approximation strategy leads us to estimate $\mathbf{\Omega}^*$ by iteratively solving a weighted graphical lasso problem with weights updated in each iteration. The proposed method is computationally efficient and can provide accurate estimation. The details of the weighted graphical lasso are explained in the next subsection.

2.3.2 Proposed Weighted Graphical Lasso

In this subsection, we will elaborate on the derivation and properties of our proposed method. First, we will introduce a lemma.

Lemma 2.3.1. *For a positive definite matrix $\mathbf{\Omega}$ with dimension p, the relationship between its determinant value and l_1 norm can be described by*

$$|\mathbf{\Omega}|^{1/2} \leq \|\mathbf{\Omega}\|_1^{p/2}.$$

The key to proving Lemma 2.3.1 is to use Gershgorin's circle theorem (Horn and Johnson 1990); see Appendix B. To overcome the nonconvexity issue in Equation (2.3.3), we first consider minimizing an upper bound of the objective function from a majorization-minimization perspective. Using Lemma 2.3.1, the minimization under consideration becomes

$$\min_{\mathbf{\Omega}}\left[-\frac{2}{n}|\mathbf{\Omega}|^{1/2}\sum_{i=1}^{n}\exp\left(-\frac{1}{2}\mathbf{x}_i^T\mathbf{\Omega}\mathbf{x}_i\right) + 2^{-p/2}\|\mathbf{\Omega}\|_1^{p/2} + \rho_n\|\mathbf{\Omega}\|_1\right].$$

Then we convert the second term above into a constraint, which takes the form $\|\mathbf{\Omega}\|_1^{p/2} \leq M_1$, and is equivalent to $\|\mathbf{\Omega}\|_1 \leq M_1^\star$, where M_1 is a large-enough constant bound and $M_1^\star = M_1^{2/p}$. This is a commonly used technique in optimization for formulating the constraints (Boyd and Vandenberghe 2004). Because $\lim_{n\to\infty}\sqrt{n}\rho_n = \rho_0$, the l_1 penalty $\rho_n\|\mathbf{\Omega}\|_1$ can also be converted into the constraint $\|\mathbf{\Omega}\|_1 \leq M_2^\star$; therefore, the minimization problem is transformed by combining the two constraints as

$$\begin{aligned}&\min_{\mathbf{\Omega}}\left[-\frac{1}{n}|\mathbf{\Omega}|^{1/2}\sum_{i=1}^{n}\exp\left(-\frac{1}{2}\mathbf{x}_i^T\mathbf{\Omega}\mathbf{x}_i\right)\right] \\ &\text{subject to} \quad \|\mathbf{\Omega}\|_1 \leq M^\star,\end{aligned} \tag{2.3.4}$$

where $M^*\,(= M_1^* + M_2^*)$ is an unknown tuning bound. This new constraint minimization illustrates the essential difference between the proposed method and the constraint likelihood-based methods. Note that the p.d.f. of the normal distribution is proportional to $|\mathbf{\Omega}|^{1/2}\exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{\Omega}\mathbf{x}\right)$ up to a constant. In Equation (2.3.4), the proposed method attempts to maximize the summation of the probabilities over the sample observations, while the likelihood-based approach attempts to maximize the product of the probabilities. As an objective function, the summation of probabilities can be more resistant to outliers than the product of those probabilities (Basu et al. 1998).

The minimization of the objective function in Equation (2.3.4) in terms of $\mathbf{\Omega}$ is equivalent to the minimization of the following objective function,

$$\min_{\mathbf{\Omega}}\left\{ -\log\left[|\mathbf{\Omega}|^{1/2}\frac{1}{n}\sum_{i=1}^{n}\exp\left(-\frac{1}{2}\mathbf{x}_i^T\mathbf{\Omega}\mathbf{x}_i\right)\right]\right\}$$

which trivially follows from the fact that the logarithm is a monotonically increasing function. Converting the constraint in Equation (2.3.4) to a penalty term in the minimization yields the objective function

$$\min_{\mathbf{\Omega}} \left\{ -\frac{1}{2}\log|\mathbf{\Omega}| - \log\left[\frac{1}{n}\sum_{i=1}^{n}\exp\left(-\frac{1}{2}\mathbf{x}_i^T\mathbf{\Omega}\mathbf{x}_i\right)\right] + \rho_n^*\|\mathbf{\Omega}\|_1 \right\}, \tag{2.3.5}$$

where ρ_n^* is another tuning parameter. In addition, it is worth pointing out that the diagonal elements of $\mathbf{\Omega}$ are penalized in the objective function Equation (2.3.5).

To further facilitate the computational advantage of the proposed method, we apply the mean-value theorem to the second term in Equation (2.3.5), denoted as $g(\mathbf{\Omega})$, with respect to $\mathbf{\Omega}$ as detailed in Appendix C. Specifically, applying the mean-value theorem to $g(\mathbf{\Omega})$ leads to

$$g(\mathbf{\Omega}) = g(\mathbf{\Omega}_0) + \frac{1}{2n}\sum_{i=1}^{n} w_i\mathbf{x}_i^T\mathbf{\Omega}\mathbf{x}_i - \frac{1}{2n}\sum_{i=1}^{n} w_i\mathbf{x}_i^T\mathbf{\Omega}_0\mathbf{x}_i,$$

where the weights w_i are defined as

$$w_i = w_i(\mathbf{\Omega}_1) \equiv \frac{\exp(-\frac{1}{2}\mathbf{x}_i^T\mathbf{\Omega}_1\mathbf{x}_i)}{\frac{1}{n}\sum_{j=1}^{n}\exp(-\frac{1}{2}\mathbf{x}_j^T\mathbf{\Omega}_1\mathbf{x}_j)} \quad \text{for } i = 1, 2, \ldots, n, \tag{2.3.6}$$

and $\mathbf{\Omega}_1$ is a point on the line segment connecting the initial estimator $\mathbf{\Omega}_0$ to the point $\mathbf{\Omega}$. Parallel to Newton's method, we use $\mathbf{\Omega}_0$ to approximate the unknown $\mathbf{\Omega}_1$ in every iteration of our algorithm. By defining $\mathbf{S}^* = \mathbf{S}^*(\mathbf{\Omega}_0) \equiv \frac{1}{n}\sum_{i=1}^{n} w_i\mathbf{x}_i\mathbf{x}_i^T$, we reformulate the objective function in (2.3.5) as

$$\min_{\mathbf{\Omega}} \Big[-\log|\mathbf{\Omega}| + \mathrm{tr}[\mathbf{\Omega}\mathbf{S}^*] + \rho_n\|\mathbf{\Omega}\|_1 \Big]. \tag{2.3.7}$$

Note that $\mathbf{S}^*$ can be viewed as a weighted sample covariance matrix where the weights $\{w_i\}$ are with respect to n observations $\mathbf{x}_1, \ldots, \mathbf{x}_n$ as well as $\mathbf{\Omega}_0$. The form of the objective function in (2.3.7) is similar to the negative log-likelihood function but with a weighted sample covariance matrix. Therefore, we refer to the solution of (2.3.7) as the *Weighted Graphical Lasso* or, more succinctly, *weighted glasso*.

Unlike the sample covariance matrix $\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T$ that assigns the same weight to each $\mathbf{x}_i$, the form of $\mathbf{S}^*$ assigns different weights to the observations $\mathbf{x}_i$. If an observation

is an outlier from the underlying distribution, its weight tends to be smaller compared with that of the average. This helps us to reduce the influence from outliers on the proposed estimate. Note that the expression in (2.3.7) depends on an initial estimator $\mathbf{\Omega}_0$ in order to calculate the weights and corresponding $\mathbf{S}^*$. We propose to iteratively minimize the objective function (2.3.7) to obtain the estimate $\widehat{\mathbf{\Omega}}$. The detailed description of the iterative algorithm is as follows.

Algorithm Weighted Glasso

---

[0] Given an initial estimate $\mathbf{\Omega_0}$ (a positive definite matrix), set the stopping threshold $\delta = 10^{-6}$.

[1] Compute w_i and obtain $\mathbf{S}^* = \frac{1}{n}\sum_{i=1}^{n} w_i \mathbf{x}_i \mathbf{x}_i^T$.

[2] Estimate $\mathbf{\Omega}^*$ by minimizing the objective function (2.3.7), i.e.,

$$\widehat{\mathbf{\Omega}} = \arg\min_{\mathbf{\Omega}} \left[-\log|\mathbf{\Omega}| + \mathrm{tr}[\mathbf{\Omega}\mathbf{S}^*] + \rho_n^* \|\mathbf{\Omega}\|_1 \right].$$

[3] Check if the Frobenius squared norm $\|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}_0\|_F^2 \leq \delta$; otherwise, set $\mathbf{\Omega}_0 = \widehat{\mathbf{\Omega}}$ and go back to Step **[1]**.

Remarks:

(1) The stopping rule uses the Frobenius squared norm (Fnorm) to measure the distance between the estimate from the $(k+1)$st iteration and that from the kth iteration,

$$\mathrm{Fnorm} \equiv \|\widehat{\mathbf{\Omega}}^{k+1} - \widehat{\mathbf{\Omega}}^{k}\|_F^2 = \sqrt{\sum_{i,j} (\widehat{c}_{ij}^{k+1} - \widehat{c}_{ij}^{k})^2}.$$

where $\widehat{c}_{ij}^{k}$ is the estimate of c_{ij} from the kth iteration.

(2) We generally take the inverse of the sample covariance matrix as the initial estimate $\mathbf{\Omega}_0$. If $p > n$, then the sample covariance matrix is singular with probability 1; hence we add a small perturbation to its diagonal elements and apply LU decomposition to derive its inverse matrix (Hsieh et al. 2011).

2.3.3 Selection of the Tuning Parameter

To select the optimal value of the tuning parameter ρ_n^*, commonly used methods include (a) cross-validation (Finegold and Drton 2011), (b) use of a validation set, and (3) information-based criteria such as the Bayesian information criterion (Yuan and Lin 2007). The cross-validation or validation-set approaches can be more appropriate than the information-based methods in situations such as ours where outliers may be present; however, traditional cross-validation may not perform well when outliers are present. When randomly partitioning the data into k folds (groups) in cross-validation, the outlier observations may not be allocated into each fold uniformly. To address this issue, we adopt a revised cross-validation method in which the observations in every fold are more likely to have a similar distribution with respect to the likelihood. Specifically, we order the observations based on the values of their likelihood functions. Suppose that there are n observations for k-fold cross-validation. The ordered observations are grouped into $\lceil n/k \rceil$ blocks, each of which contains k observations. Then every fold for cross validations is formed by randomly drawing one observation from each block without replacement. In this manner, the data points in each fold tend to be uniformly distributed based on the likelihood values with respect to $\mathbf{\Omega_0}$. To obtain the cross-validation score, the values of ISE in (2.3.2) are computed for every hold-out set in cross-validation. We select the optimal ρ_n corresponding to the smallest ISE value.

2.4 Simulation Experiments

2.4.1 Numerical Simulation

To assess the performance of the proposed method, we conduct a set of simulations to compare the proposed method with the LW, glasso, and tlasso methods introduced in Section 2.2 for estimating the inverse covariance matrix.

For the simulated data, we consider samples of size n_1 that are generated from a multivariate normal $N_p(\mathbf{0}, \mathbf{\Sigma})$ but are contaminated by n_2 outlier observations generated from

a $N_p(\boldsymbol{\mu}, \boldsymbol{I})$. That is, the total sample size of the data set is $n = n_1 + n_2$. Here we fix $n_1 = 50$ and vary the outlier-to-signal ratios $\gamma \equiv \frac{n_2}{n_1} = 0\%$, 6%, and 10%. The dimensionality choices are $p = 55$ and 100. Moreover, we also investigate the performance of the proposed method under two different scenarios of mean shift vectors $\boldsymbol{\mu} = \mathbf{2}$ and $\mathbf{5}$. Three inverse covariance models $\boldsymbol{\Omega}$ are considered as follows.

Model 1. Identity matrix, i.e., $\boldsymbol{\Omega} = \mathbf{I}$. Denote by M1-I and M1-II the model under mean shifts of $\boldsymbol{\mu} = \mathbf{2}$ and $\boldsymbol{\mu} = \mathbf{5}$, respectively.

Model 2. Banded-structured matrix, i.e., $\boldsymbol{\Omega} = (c_{ij})$ with $c_{ii} = 1$, $c_{i+1,i} = c_{i,i+1} = c$, $c_{ij} = 0$, for $|i - j| > 1$. The corresponding covariance matrix $\boldsymbol{\Sigma} = (e_{ij})$ with

$$e_{ii} = \frac{\beta - \alpha}{(\beta^{n-1} - \alpha^{n-1})(1 - c^2) - (\beta^{n-2} - \alpha^{n-2})c^2},$$

$$e_{i+1,i} = e_{i,i+1} = \frac{-(\beta - \alpha)c}{(\beta^{n-1} - \alpha^{n-1})(1 - c^2) - (\beta^{n-2} - \alpha^{n-2})c^2},$$

and

$$e_{ij} = 0, \text{for } |i - j| > 1, \text{where } c = 0.2, \alpha = \tfrac{1-\sqrt{1-4c^2}}{2}, \text{and } \beta = \tfrac{1+\sqrt{1-4c^2}}{2}.$$

Denote by M2-I and M2-II the model under mean shifts of $\boldsymbol{\mu} = \mathbf{2}$ and $\boldsymbol{\mu} = \mathbf{5}$, respectively.

Model 3. Randomly permuted banded-structured matrix, i.e., $\boldsymbol{\Omega} = \mathbf{Q}\mathbf{M}\mathbf{Q}^T$, where $\mathbf{M}$ is the inverse covariance matrix defined in Model 2, and $\mathbf{Q}$ is a $p \times p$ matrix obtained by randomly permuting the rows of a $p \times p$ identity matrix. Denote by M3-I and M3-II the model under mean shifts of $\boldsymbol{\mu} = \mathbf{2}$ and $\boldsymbol{\mu} = \mathbf{5}$, respectively.

To evaluate the accuracy of the estimated covariance matrix, we use the F_1 score, Frobenius norm, and Kullback-Leibler (KL) loss as three performance criteria. The F_1 score, a measure of selection accuracy, is defined as

$$F_1 = \frac{2PR}{P + R},$$

where $P = tp/(tp + fp)$ is the precision and $R = tp/(tp + fn)$ is the recall. Here, tp is the number of true positives, fp is the number of false positives, and fn is the number of

false negatives (Davis and Goadrich 2006). In our problem setting, tp indicates the number of correct cases in which both the estimated entries of $\widehat{\mathbf{\Omega}}$ and the corresponding entries of $\mathbf{\Omega}^*$ are nonzeros; fp indicates the number of estimated entries incorrectly identified to be nonzeros while the associated true values are zeros; and the fn is contrary to fp in that it counts all the misidentifications in which the estimated entries are zeros but the true ones are nonzeros. The KL loss is a likelihood-based measurement for the accuracy of the estimate, and is defined as

$$\mathrm{KL} \equiv -\log(|\mathbf{\Sigma}^{-1}|) + \mathrm{tr}(\widehat{\mathbf{\Sigma}}^{-1}\mathbf{\Sigma}) + \log(|\widehat{\mathbf{\Sigma}}^{-1}|) - p.$$

The larger the F_1 score, the more accurate the estimate; but smaller Frobenius norm and KL loss indicate better estimation performance.

The numerical results are shown in Tables 1–3 and are based on 100 simulation experiments. The standard deviations are in parentheses. Table 1 considers the situation of non-outlier observations. Note that the glasso method is expected to perform best under the non-outlier case. We see that the performance of our proposed method is comparable to the glasso method and better than the tlasso method. Table 2 shows the results for the 6% outlier-to-signal ratio case. When the dimension $p = 55$ and the mean shift $\boldsymbol{\mu} = \mathbf{2}$, the proposed method has comparable performance to the LW method in terms of KL loss for Models 1 and 2. However, the F_1 score of the proposed method is much larger than that of the LW method, indicating its superiority in terms of selection accuracy. When the dimension p increases, as in the case of $p = 100$, the proposed method obtains larger F_1 scores and smaller Fnorm and KL loss values than the other three methods. As an example for which anomalous observations become more significant, the results of the 10% outlier-to-signal ratio case are reported in Table 3. In this case, the proposed method always yields the largest F_1 score and the smallest KL loss among the four methods under comparison. The values of Fnorm for the proposed method are close to those of the glasso method, and much better than those of the other two methods. Overall, these simulation results show that the proposed method generally outperforms the other three methods when outliers are

present.

Table 1: Simulation results under the case of no outliers

		Model 1			Model 2			Model 3		
p	Method	F_1	$\|\cdot\|_F$	KL	F_1	$\|\cdot\|_F$	KL	F_1	$\|\cdot\|_F$	KL
	weighted glasso	0.31	8.52	6.11	0.31	8.59	6.13	0.38	4.03	3.81
		(0.04)	(0.62)	(0.35)	(0.03)	(0.66)	(0.39)	(0.07)	(0.64)	(0.40)
	glasso	0.32	7.16	3.94	0.33	7.48	4.04	0.27	3.24	2.04
		(0.07)	(0.92)	(0.56)	(0.07)	(1.41)	(0.80)	(0.11)	(0.84)	(0.45)
55	tlasso	0.11	25.50	35.76	0.12	25.54	35.87	0.05	23.22	35.26
		(0.00)	(0.46)	(0.82)	(0.00)	(0.53)	(0.93)	(0.00)	(0.36)	(0.82)
	LW	0.10	4.30	2.34	0.10	4.31	2.35	0.04	0.27	0.13
		(0.00)	(0.19)	(0.11)	(0.00)	(0.16)	(0.11)	(0.00)	(0.23)	(0.11)
	weighted glasso	0.24	15.96	10.66	0.25	16.04	10.70	0.30	7.06	6.62
		(0.04)	(1.42)	(0.79)	(0.04)	(1.12)	(0.64)	(0.06)	(0.92)	(0.81)
	glasso	0.29	14.03	7.46	0.29	13.95	7.48	0.29	5.76	3.80
		(0.06)	(1.23)	(0.56)	(0.07)	(1.12)	(0.49)	(0.10)	(0.75)	(0.52)
100	tlasso	0.07	50.12	113.11	0.07	50.24	113.33	0.03	46.67	111.70
		(0.00)	(0.64)	(1.11)	(0.00)	(0.75)	(1.16)	(0.00)	(0.28)	(1.27)
	LW	0.06	8.15	4.42	0.06	8.15	4.42	0.06	0.43	0.22
		(0.00)	(0.23)	(0.16)	(0.00)	(0.24)	(0.16)	(0.19)	(0.33)	(0.16)

2.4.2 Genetic Network Simulation

In the genetic study of Section 2.5.1, graphical models are applied to infer the genetic network so as to study the conditional correlations among various genes. Of course, it is hard to verify the correctness of the inferred network because of the lack of the true biological information. In this section, we mimic the genetic study and set up a simulation experiment with a "known" genetic network. We will compare the estimates from the weighted glasso, glasso, and tlasso methods with the true state of affairs.

We generate 20 genes named 1 to 20 with known conditional correlations. The true graphical model consists of two parts: a cluster among genes 1–4, and a hub structure with gene 10. The corresponding inverse covariance matrix is denoted as $\boldsymbol{D}$, and we sample 20 observations from the multivariate normal distribution $N_{20}(\mathbf{0}, \boldsymbol{D})$ and 2 outliers from the outlier distribution $N_{20}(\mathbf{0}, \boldsymbol{I})$.

Figure 1 shows the true graphical model and the estimates from the weighted glasso, glasso, and tlasso methods. We can see that the weighted glasso gives the most-sparse

Table 2: Simulation results for the outlier-to-signal ratio 6% case

		M1-I			M2-I			M3-I		
p	Method	F_1	$\|\cdot\|_F$	KL	F_1	$\|\cdot\|_F$	KL	F_1	$\|\cdot\|_F$	KL
	weighted glasso	0.31	8.78	6.26	0.31	8.72	6.24	0.39	4.10	3.92
		(0.03)	(1.10)	(0.58)	(0.04)	(0.85)	(0.49)	(0.06)	(0.76)	(0.48)
	glasso	0.16	10.21	6.62	0.16	9.93	6.41	0.12	5.61	4.43
		(0.01)	(1.37)	(0.80)	(0.01)	(0.77)	(0.52)	(0.01)	(1.21)	(0.51)
55	tlasso	0.12	26.47	37.17	0.12	26.50	37.08	0.12	23.52	36.34
		(0.00)	(0.41)	(0.79)	(0.00)	(0.40)	(0.79)	(0.00)	(0.33)	(0.76)
	LW	0.10	8.81	5.76	0.10	8.80	5.74	0.04	8.19	4.89
		(0.00)	(0.76)	(0.51)	(0.00)	(0.77)	(0.50)	(0.00)	(1.63)	(0.68)
	weighted glasso	0.25	16.10	10.80	0.25	16.20	10.80	0.30	7.10	6.50
		(0.04)	(1.27)	(0.68)	(0.04)	(1.31)	(0.73)	(0.06)	(1.36)	(0.77)
	glasso	0.12	17.90	11.41	0.12	17.83	11.38	0.10	9.21	7.52
		(0.01)	(0.86)	(0.36)	(0.01)	(0.88)	(0.35)	(0.01)	(1.25)	(0.56)
100	tlasso	0.07	51.89	115.54	0.07	51.54	115.37	0.025	47.12	113.60
		(0.00)	(4.33)	(2.08)	(0.00)	(0.44)	(1.10)	(0.00)	(0.31)	(1.28)
	LW	0.06	20.95	14.06	0.06	20.75	13.92	0.02	23.14	12.55
		(0.00)	(1.91)	(1.03)	(0.00)	(2.10)	(1.20)	(0.00)	(3.33)	(1.18)
		M1-II			M2-II			M3-II		
		F_1	$\|\cdot\|_F$	KL	F_1	$\|\cdot\|_F$	KL	F_1	$\|\cdot\|_F$	KL
	weighted glasso	0.30	9.67	6.86	0.30	9.69	6.87	0.37	4.49	4.50
		(0.03)	(1.15)	(0.76)	(0.04)	(1.09)	(0.71)	(0.07)	(0.78)	(0.74)
	glasso	0.13	10.95	8.98	0.13	10.74	8.82	0.10	6.44	6.76
		(0.02)	(1.27)	(0.66)	(0.02)	(0.75)	(0.42)	(0.01)	(1.35)	(0.50)
55	tlasso	0.12	26.63	38.94	0.11	26.58	38.94	0.12	23.53	37.91
		(0.00)	(0.40)	(0.76)	(0.00)	(0.40)	(0.81)	(0.00)	(0.36)	(0.82)
	LW	0.10	11.28	9.83	0.10	11.25	9.79	0.04	8.64	8.56
		(0.00)	(0.25)	(0.18)	(0.00)	(0.30)	(0.22)	(0.00)	(0.20)	(0.19)
	weighted glasso	0.25	17.60	11.60	0.25	17.94	11.90	0.29	8.02	7.49
		(0.03)	(1.98)	(52.38)	(0.04)	(2.10)	(1.30)	(0.06)	(1.40)	(1.21)
	glasso	0.10	18.73	14.14	0.10	18.91	14.19	0.09	10.15	10.18
		(0.01)	(1.00)	(0.58)	(0.01)	(1.00)	(0.50)	(0.01)	(1.30)	(0.59)
100	tlasso	0.07	51.60	117.09	0.07	51.56	117.34	0.03	47.16	115.50
		(0.00)	(0.67)	(1.24)	(0.00)	(0.47)	(1.07)	(0.00)	(0.25)	(1.10)
	LW	0.06	23.60	19.90	0.06	23.55	19.90	0.02	20.23	17.14
		(0.00)	(0.57)	(0.22)	(0.00)	(0.56)	(0.22)	(0.00)	(0.73)	(0.22)

Table 3: Simulation results for the outlier-to-signal ratio 10% case

		M1-I			M2-I			M3-I		
p	Method	F_1	$\|\cdot\|_F$	KL	F_1	$\|\cdot\|_F$	KL	F_1	$\|\cdot\|_F$	KL
	weighted glasso	0.30	9.01	6.42	0.31	9.03	6.46	0.35	4.44	4.27
		(0.03)	(0.84)	(0.53)	(0.03)	(0.88)	(0.55)	(0.06)	(1.28)	(0.86)
	glasso	0.14	10.44	7.12	0.14	10.27	7.05	0.11	5.99	5.06
		(0.01)	(1.18)	(0.64)	(0.01)	(0.64)	(0.36)	(0.02)	(1.73)	(0.65)
55	tlasso	0.11	26.27	35.23	0.11	28.62	35.54	0.04	26.32	34.63
		(0.00)	(0.49)	(0.73)	(0.00)	(22.98)	(2.73)	(0.00)	(22.56)	(2.43)
	LW	0.10	21.49	11.61	0.10	21.34	11.57	0.04	27.88	11.65
		(0.00)	(2.76)	(0.87)	(0.00)	(3.16)	(1.06)	(0.00)	(4.56)	(1.10)
	weighted glasso	0.27	11.27	8.12	0.26	11.71	8.45	0.25	6.46	6.47
		(0.03)	(1.84)	(1.43)	(0.03)	(2.21)	(1.98)	(0.05)	(2.22)	(2.52)
	glasso	0.13	10.75	9.26	0.13	10.67	9.24	0.10	6.39	7.19
		(0.02)	(0.90)	(0.46)	(0.01)	(0.60)	(0.33)	(0.01)	(1.52)	(0.54)
100	tlasso	0.12	26.61	38.62	0.11	26.35	37.02	0.04	26.56	36.51
		(0.00)	(0.45)	(0.76)	(0.00)	(0.52)	(0.77)	(0.00)	(24.37)	(2.66)
	LW	0.10	18.23	13.51	0.10	18.24	13.52	0.04	17.73	12.26
		(0.00)	(0.82)	(0.28)	(0.00)	(0.79)	(0.28)	(0.00)	(0.86)	(0.22)
		M1-II			M2-II			M3-II		
		F_1	$\|\cdot\|_F$	KL	F_1	$\|\cdot\|_F$	KL	F_1	$\|\cdot\|_F$	KL
	weighted glasso	0.23	20.99	13.95	0.24	16.69	11.11	0.29	7.47	6.95
		(0.03)	(3.10)	(2.09)	(0.04)	(1.72)	(0.98)	(0.03)	(1.05)	(0.95)
	glasso	0.10	19.21	14.76	0.11	18.33	12.21	0.09	9.58	8.37
		(0.01)	(3.17)	(1.55)	(0.01)	(1.08)	(0.56)	(0.01)	(0.87)	(0.56)
55	tlasso	0.07	53.04	115.42	0.07	52.31	115.14	0.03	47.54	113.28
		(0.00)	(4.96)	(2.20)	(0.00)	(0.43)	(1.08)	(0.00)	(0.27)	(1.16)
	LW	0.06	54.61	32.67	0.06	68.18	32.22	0.02	93.64	33.23
		(0.00)	(2.18)	(0.52)	(0.00)	(8.17)	(2.40)	(0.00)	(11.11)	(2.40)
	weighted glasso	0.24	21.23	14.06	0.23	21.10	14.02	0.26	10.91	10.06
		(0.03)	(2.69)	(1.86)	(0.03)	(3.13)	(2.09)	(0.03)	(2.03)	(1.68)
	glasso	0.10	18.87	14.59	0.10	19.16	14.73	0.09	9.99	10.55
		(0.01)	(1.14)	(0.64)	(0.01)	(3.20)	(1.55)	(0.01)	(0.90)	(0.55)
100	tlasso	0.07	52.39	117.02	0.07	52.39	116.90	0.03	47.61	115.21
		(0.00)	(0.58)	(1.07)	(0.00)	(0.58)	(1.20)	(0.00)	(0.30)	(1.20)
	LW	0.06	54.46	32.63	0.06	54.62	32.67	0.02	56.82	29.25
		(0.00)	(2.09)	(0.49)	(0.00)	(2.18)	(0.52)	(0.00)	(2.40)	(0.44)

results and is the closest to the truth; the glasso and tlasso methods give less-sparse estimates and identify some genes and conditional correlations that do not appear in the true graphical model.

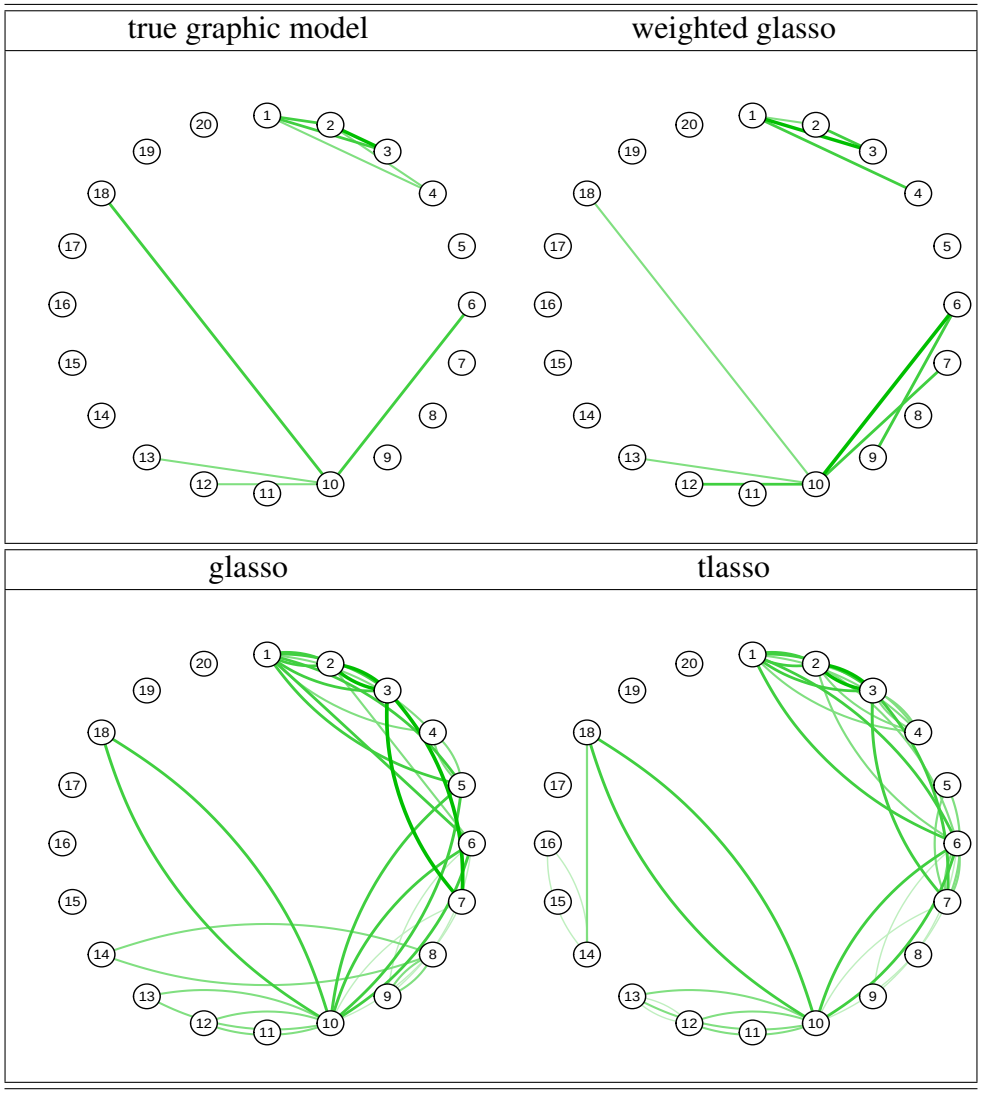


Figure 1: Graphical models estimation of three methods compared to the true graphical model

2.4.3 Monte Carlo Simulations for Convergence

The weighted glasso objective function (2.3.7) is nonconvex and nonsmooth, and the weights involved are a complicated function of the observations and the inverse covariance matrix, so it is difficult to show the convergence of the estimate theoretically. We will

use Monte Carlo simulations to investigate our algorithm's convergence performance. In the meantime, we will compare it with the glasso method, whose objective function is convex and known to converge. We utilize our previous three numerical simulation settings in which the true inverse covariance matrices are the identity matrix, banded-structured matrix, and random banded-structured matrix. We consider a lower dimension $p = 55$ and a higher dimension $p = 100$ under three outlier-to-signal ratios $0\%, 6\%$ and 10%, where the mean of the outliers is $\boldsymbol{\mu} = \mathbf{5}$.

To study the performance of the two methods as the total sample size n increases, we enlarge the sample size n_1 while keeping the outlier-to-signal ratio unchanged for the specific case. In the asymptotic situation, the sample size n should be much larger than the dimension p. Here we assume that the largest total sample size n is roughly of the scale p^2. In particular, for the lower dimension $p = 55$, we pick 16 values of n_1 ranging up to 2000; for the higher dimension $p = 100$, the value of n_1 ranges from 30 to 10,000. We assess the distance between the estimate and the true inverse covariance matrix by the Frobenius squared norm. All the results are based on 100 independent replications.

Figures 2–4 depict the convergence performance results of the weighted glasso and the glasso methods under the three numerical models, respectively. We can see that as the total sample size increases, the Fnorm distances of the two estimates from the true inverse covariance matrix decrease gradually and become stable in the later stages. Under the no-outlier situation, the glasso method yields a smaller distance to the true inverse covariance matrix than the weighted glasso does. In the cases of positive outlier-to-signal ratios, the average distance between the weighted glasso estimate and the true inverse covariance matrix is smaller than that of the glasso estimate. This suggests that the weighted glasso estimate converges to a better position in the neighborhood of the true inverse covariance matrix when the data are noisy. The glasso estimate has a sharper decrease in the Fnorm value than the weighted glasso estimate, especially at the beginning stages where the total sample size n is much smaller than the dimension p. This illustrates the robustness of the

weighted glasso in a high-dimensional situation.

From the above Monte Carlo simulations, it appears that the weighted glasso estimate converges a bit more slowly to the true inverse covariance matrix, and has a smaller Fnorm distance to the true one when outliers are present, compared to the glasso estimate.

2.5 Case Studies

In this section, we illustrate the performance of the proposed method through two real-world applications: (1) gene network inference for breast cancer data, and (2) portfolio optimization using Standard & Poor's industrial index (SP100) data.

2.5.1 Gene Network of Breast Cancer Data

The Gaussian graphical model is widely used in gene network applications for identifying important relationships among genes. Here we adopt the gene expression data provided by Hess et al. (2006) to perform inference on genetic networks. This data set contains 133 patients with stage I–III breast cancer, who were treated with chemotherapy. Two clusters are defined based on the patient responses to the treatment: pathologic complete response (pCR) and residual disease (not-pCR). Hess et al. (2006) and Natowicz et al. (2008) identified 26 key genes important for the treatment. The detailed description of these genes is listed in Appendix D. As suggested by Ambroise et al. (2009), cases from the two classes pCR (34 patients) and not-pCR (99 patients) do not have the same distribution since they were treated under different experimental conditions. Thus we apply the proposed Gaussian graphical model procedure to each cluster separately. We compared the proposed method for estimating $\mathbf{\Sigma}^{-1}$ with the glasso and tlasso methods. Here the LW method is not included for comparison since it would not produce a sparse graph for the gene network. The relationship between the estimate and the gene network arises in that an entry $\widehat{c}_{i,j} \neq 0$ means that the ith and jth genes are conditionally dependent; otherwise, they are conditionally independent given the rest of the genes. The most-dependent genes indicate their potential significance for breast cancer and will be emphasized in a later genetic study.

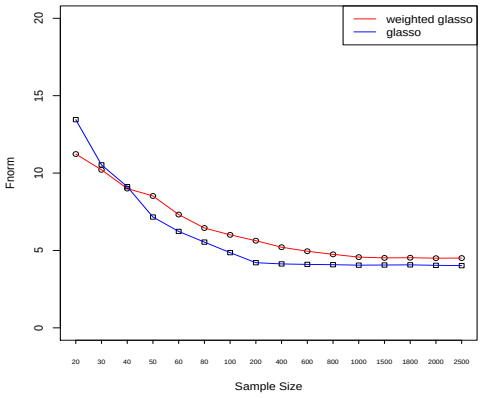
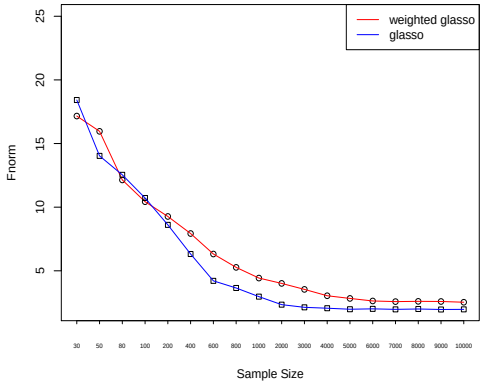
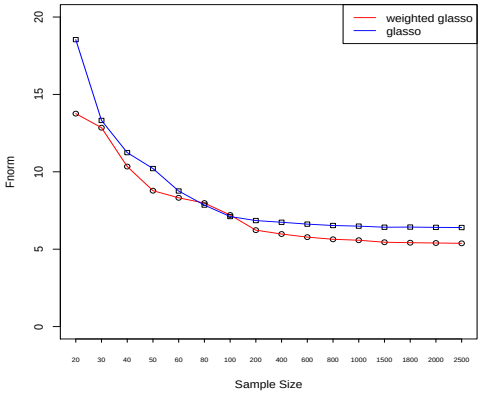
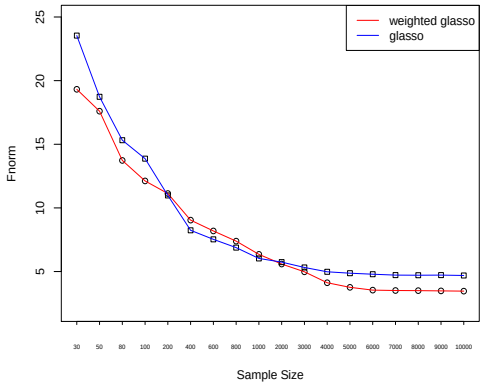
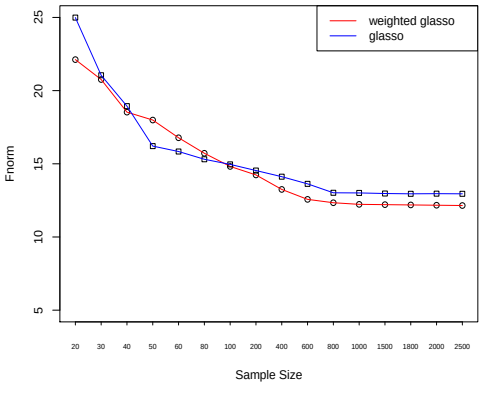
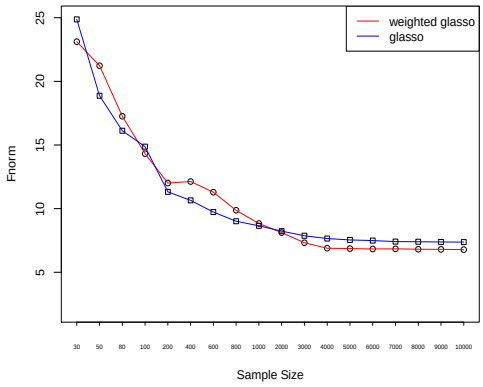
outlier-to-signal ratio	$p = 55$	$p = 100$
0%		
6%		
10%		

Figure 2: Convergence of the weighted glasso and glasso under the Model 1 setting

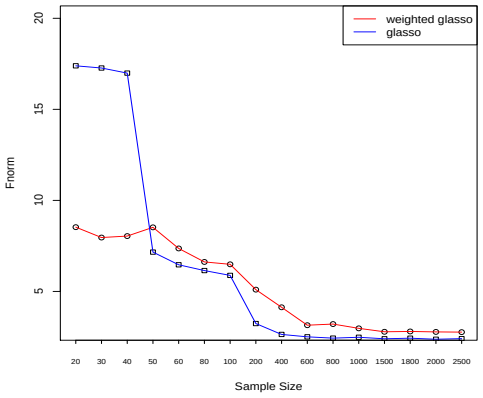
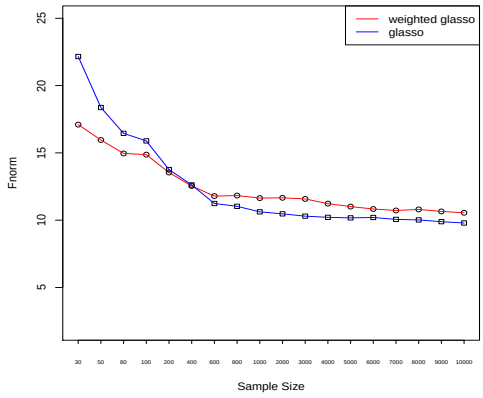
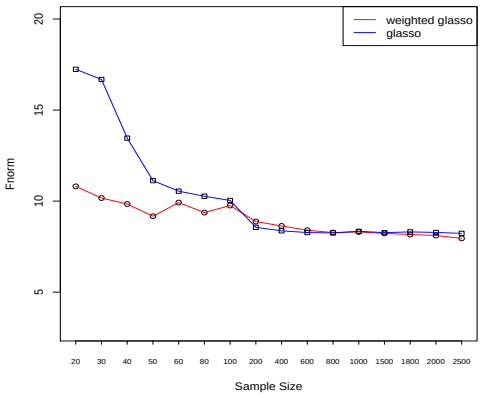
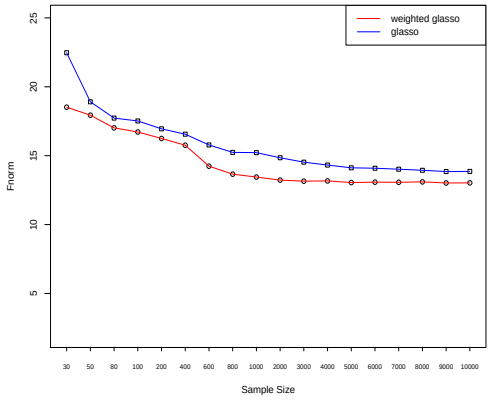
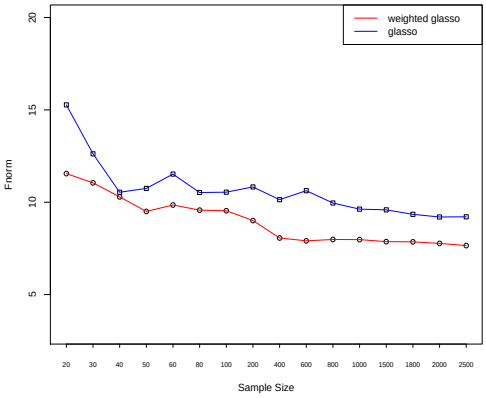
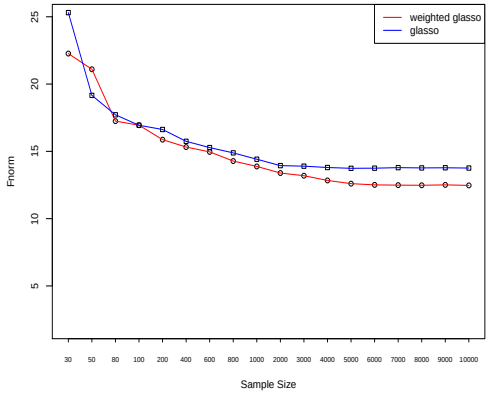
outlier-to-signal ratio	$p = 55$	$p = 100$
0%		
6%		
10%		

Figure 3: Convergence of the weighted glasso and glasso under the Model 2 setting

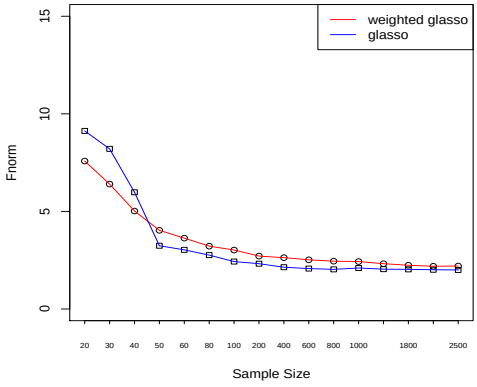
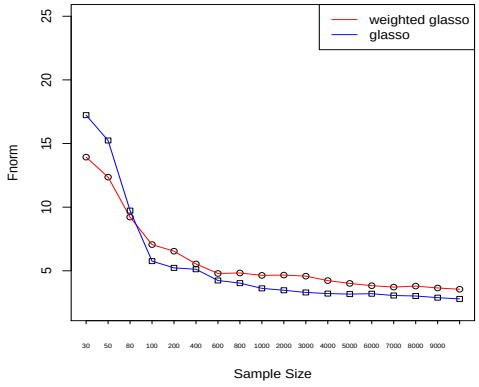
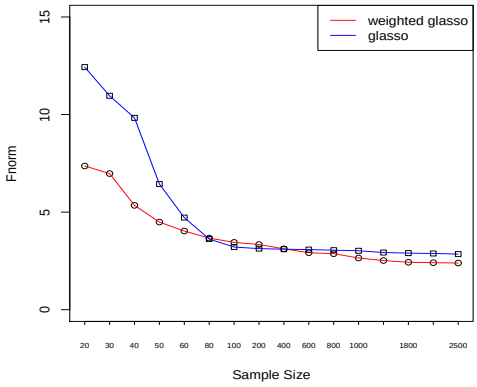
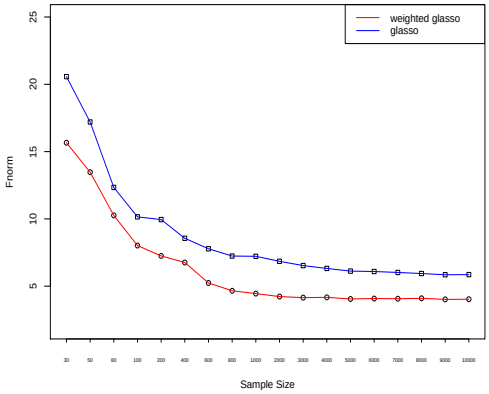
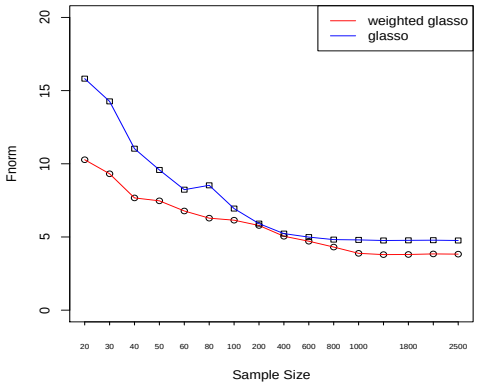
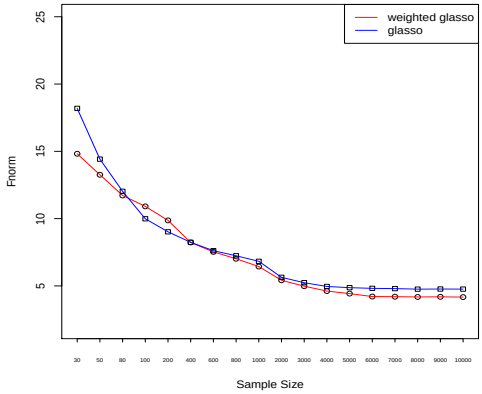
outlier-to-signal ratio	$p = 55$	$p = 100$
0%		
6%		
10%		

Figure 4: Convergence of the weighted glasso and glasso under the Model 3 setting

Figure 5 shows the inferred networks of the three methods for studying pCR and not-pCR classes. The penalty values are obtained by the revised cross-validation in Section 2.3.3. For the results from the proposed weighted glasso method, the structure of the not-pCR network is more centered than that of the pCR network. All the edges in the network from not-pCR connect with the gene SCUBE2, which implies its significant role in the study. The network for the pCR case involves more genes by which we can discover the pairwise relations among BTG3, METRN, and MELK, and among BTG3, SCUBE2, and IGFBP4, to name a few. Our findings confirm the results in Ambroise et al. (2009), who also identified SCUBE2 as a hub gene for the not-pCR class and the cluster structure for the pCR class. In contrast, results from the glasso and the tlasso methods tend to give less-sparse structures, where they identify more gene connections in either the pCR or not-pCR networks.

To further elaborate on the proposed method for exploring the structure of the gene network, we report in Figure 6 the estimated gene networks under low, middle, and high-level values of the tuning parameter ρ_n^* for pCR and not-pCR. For the pCR cluster, gene BTG3 keeps its importance to the network for all levels of penalty values. This consistent property further validates the significance of BTG3, as indicated in Ambroise et al. (2009). For the not-pCR cluster, we see that genes MAPT, METRN, and MELK tend to be more significant in the low and middle penalized networks, and gene SCUBE2 is identified as a key one in the high-level network. These results may provide additional information on potentially important genes when further investigating the network structures of the two classes.

2.5.2 Portfolio Optimization of SP100 Index Data

We apply the proposed weighted glasso method in the application of portfolio optimization. Markowitz (1952) developed the mean-variance optimization approach, where the risk of a portfolio $\mathbf{w} = (w_1, \ldots, w_p)$ is often measured by the standard deviation $\sqrt{\mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w}}$ of

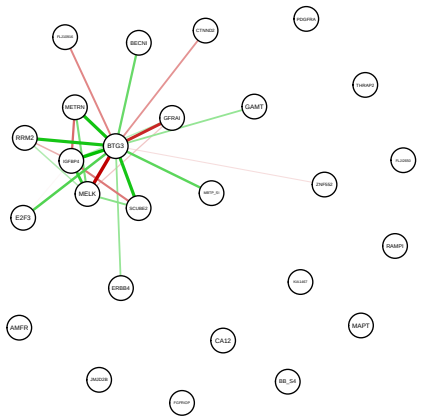
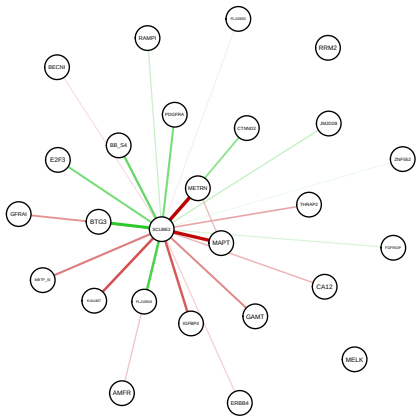
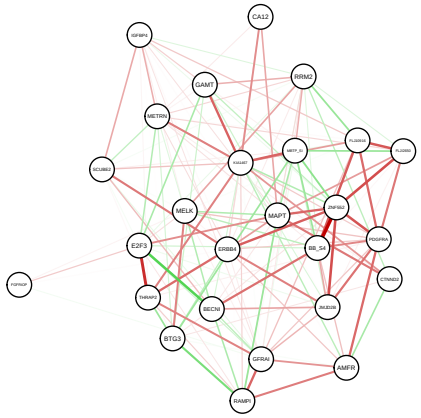
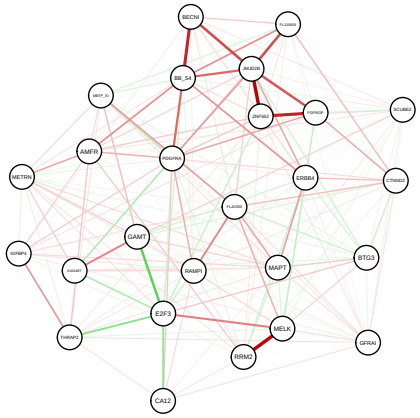
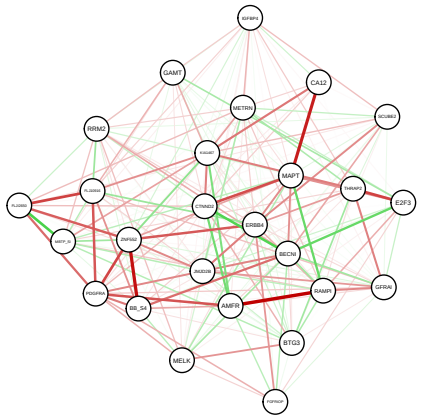
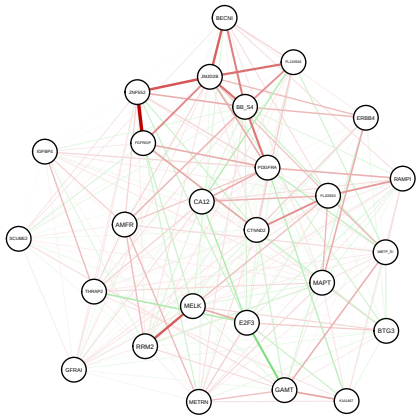
	pcR class	not-pcR class
weighted glasso		
glasso		
tlasso		

Figure 5: Results of three methods in breast cancer gene expressions

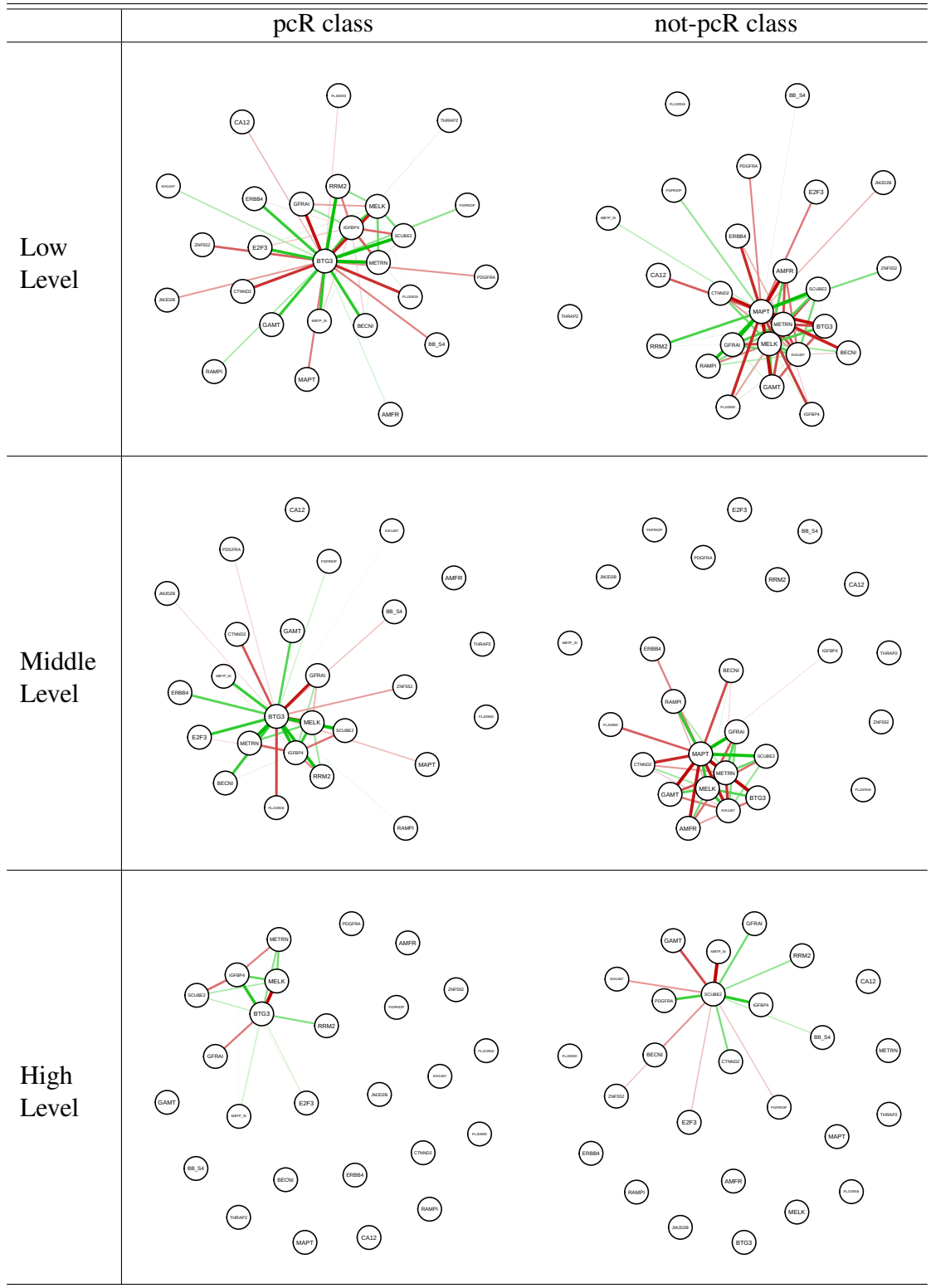


Figure 6: Results of weighted glasso in breast cancer gene expressions under different levels of penalty values

its return, where $w_i \geq 0$ and $\sum_{i=1}^{p} w_i = 1$. The estimated minimum-variance portfolio optimization problem can be expressed as

$$\min_{\mathbf{w}} \mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w} \tag{2.5.1}$$
$$\text{subject to} \sum_{i=1}^{p} w_i = 1,$$

where $\boldsymbol{\Sigma}$ is the covariance matrix of the stock prices. The solution of the constrained minimization problem involves the use of the inverse covariance matrix in that the estimator for $\mathbf{w}$ is $\widehat{\mathbf{w}} = (\widehat{\boldsymbol{\Omega}}\mathbf{1})/(\mathbf{1}'\widehat{\boldsymbol{\Omega}}\mathbf{1})$. Since the assignment of weights is crucial to the portfolio, we expect that a more-accurate estimate of the inverse covariance matrix will yield a more-precise solution and thus further a better portfolio scheme. Of course, several financial crises have occurred in the past decade, so the U.S. stock-price data are very likely to be contaminated with anomalous observations. To overcome this challenge, robustness of the inverse covariance matrix estimation is essential when conducting the above portfolio optimization.

To elaborate on the benefits arising from the robustness of the proposed method, we consider the weekly returns of the 100 components from the SP100 index from 2007 to 2010. By removing companies not in the SP100 during the financial crisis, we get 98 components out of the 100 stocks. The data consist of adjusted closing prices of the weekly returns and were extracted from Yahoo! Finance (`http://finance.yahoo.com`). Having the data for four years, we consider two time periods: 2007–2009 and 2008–2010. During each period, we use the data from the first two years as the training set and the data from the last year as the test set. The revised cross-validation in Section 2.3.3 is used to select the tuning parameters.

For the evaluation of the estimated portfolio $\mathbf{w}$, we measure the realized return, the realized risk, the realized ratio, and the Sharpe ratio from the test set. Denote $\mathcal{X}_{\text{test}}$ as the test set and $\mathbf{S}_{\text{test}}$ as the sample covariance matrix of $\mathcal{X}_{\text{test}}$. The realized return is defined as

$$R(\mathbf{w}) = \sum_{\mathbf{x} \in \mathcal{X}_{\text{test}}} \mathbf{w}^T \mathbf{x},$$

Table 4: Comparison of realized return, risk, and ratios. The tuning parameters are chosen based on cross-validation.

	2007–2009				2008–2010			
	return	risk	realized ratio	Sharpe ratio	return	risk	realized ratio	Sharpe ratio
weighted glasso	0.339	0.041	8.340	0.157	0.175	0.024	7.177	0.133
glasso	0.311	0.037	8.308	0.156	0.164	0.023	7.138	0.132
tlasso	0.086	0.021	3.757	0.068	0.067	0.015	4.504	0.080
LW	−0.118	0.020	−5.984	−0.118	0.109	0.015	7.066	0.129

and the realized risk is

$$\sigma(\mathbf{w}) = \sqrt{\mathbf{w}^T \mathbf{S}_{\text{test}} \mathbf{w}}.$$

Correspondingly, the realized ratio is

$$t(\mathbf{w}) = \frac{R(\mathbf{w})}{\sigma(\mathbf{w})} = \frac{\sum_{\mathbf{x} \in \mathcal{X}_{\text{test}}} \mathbf{w}^T \mathbf{x}}{\sqrt{\mathbf{w}^T \mathbf{S}_{\text{test}} \mathbf{w}}}.$$

We also employ the Sharpe ratio as another comparison criterion. The Sharpe ratio characterizes how well the return of an asset compensates the risk taken by the investor (Sharpe 1994). Let $\mathcal{X}_j$ be the jth weekly stock price data set; then the jth weekly return $r_j(\mathbf{w}) = \sum_{\mathbf{x} \in \mathcal{X}_j} \mathbf{w}^T \mathbf{x}$. We define the Sharpe ratio $s(\mathbf{w})$ on the weekly realized returns as

$$s(\mathbf{w}) = \frac{\mathrm{E}[r_j(\mathbf{w})] - r_f}{\sqrt{\mathrm{Var}[r_j(\mathbf{w})]}}, \tag{2.5.2}$$

where r_f is the risk-free interest rate. As in (2.5.2), the Sharpe ratio determines the reward per unit of risk by subtracting the risk-free rate from the realized returns of a portfolio scaled by the standard deviation of the returns. Clearly, the higher the Sharpe ratio, the better the portfolio's risk-adjusted performance.

The estimate of the optimal portfolio assignment $\mathbf{w}$ is computed with $\mathbf{\Sigma}^{-1}$ estimated by the weighted glasso, glasso, tlasso, and LW methods, separately. The realized returns, realized risks, realized ratios, and the Sharpe ratios for the four methods are reported in Table 4. Among the four methods, the portfolio from the proposed method gave larger realized returns $R(\mathbf{w})$, realized ratios $t(\mathbf{w})$, and Sharpe ratios $s(\mathbf{w})$ than the portfolios from the other three methods. The realized risk using the proposed method is comparable

to that using the glasso method. It appears that the proposed method improves the portfolio scheme with a better weight allocation.

Since stock-price data have a temporal order in nature, we also evaluate the proposed method using the validation set approach for selecting the tuning parameters. Specifically, for each session, we divide the three-year data into a training set, validation set, and test set associated with each year. The results shown in Table 5 further confirm that the proposed method gives the best realized return, realized ratio, and Sharpe ratio among the four methods in the comparison. This finding is also consistent with the results from using cross-validation for selecting the tuning parameters.

Table 5: Comparison of realized return, risk, and ratios. The tuning parameters are chosen based on a validation set.

	2007–2009				2008–2010			
	return	risk	realized ratio	Sharpe ratio	return	risk	realized ratio	Sharpe ratio
weighted glasso	0.340	0.041	8.362	0.157	0.174	0.024	7.182	0.133
glasso	0.333	0.040	8.264	0.155	0.161	0.023	7.029	0.130
tlasso	0.271	0.038	7.228	0.135	0.063	0.015	4.153	0.073
LW	0.031	0.027	1.149	0.019	0.084	0.016	5.241	0.094

2.6 Discussion and Conclusions

In this chapter, we studied robust estimation methods for the inverse of a covariance matrix. The proposed method employed the nonparametric ISE criterion as a loss function for a parametric model, which led to a robust estimator having a sparse structure. Adopting a majorization-minimization technique, we solved a relaxed optimization problem and developed an iteratively weighted graphical lasso estimation procedure. Under such a formulation, the weights can be updated automatically in each iteration, which effectively reduces the influence of potential anomalous observations. This opens up a new possibility to better infer graphical models.

The results of a simulation study showed that the proposed estimator is more robust to anomalous observations compared to conventional methods. We also conducted a Monte

Carlo simulation to assess the convergence properties of the algorithm. In terms of computational efficiency, our simulation experience is that the computation time of the proposed algorithm is similar to that of the glasso method, but much faster than the tlasso method. Note that the tlasso method appears to require more computational time because of its use of the EM algorithm.

We used an l_1 penalty on the objective function to obtain the sparse structure of the graphical model. Other penalties can also be considered into the proposed method, such as the elastic net (Zou and Hastie 2005), which is a mixture of l_1 and l_2 penalties. However, it is worth pointing out that the proposed method has taken advantage of the l_1 penalty for absorbing the upper bound of the objective function in (2.3.4). The effective accommodation of other types of penalties without extra computational cost is one of our future research topics.

CHAPTER III

SEQUENTIAL PROCEDURES FOR ESTIMATING THE STEADY-STATE MEAN USING STANDARDIZED TIME SERIES

3.1 Introduction

Computer simulation is perhaps the most widely used tool in the fields of industrial engineering, operations research, and the management sciences. Steady-state simulations play a fundamental role in system design, and they are particularly appropriate for evaluating long-run system performance or risk. For instance, what is the steady-state expected return from a certain financial management strategy, or what is the long-term probability of a default? Although there are now many commercial and public-domain software packages supporting the development of valid and efficient simulation models for complex systems, rarely have these packages been equipped with *comprehensive* facilities for performing rigorous, state-of-the-art statistical analysis of the outputs arising from steady-state simulation experiments. In many large-scale simulation applications, most of the effort is devoted to the development and execution of computer-based models, while relatively little attention is devoted to careful follow-up analysis of the final results.

A fundamental problem in simulation output analysis concerns the computation of point and confidence interval (CI) estimators for the mean μ of a stationary discrete-time stochastic process $\boldsymbol{X} \equiv \{X_i : i = 1, 2, \ldots\}$. The point estimation of μ is actually a relatively easy problem when the underlying system starts in steady state; based on a simulation-generated time series $\{X_1, \ldots, X_n\}$ of length n, the sample mean $\overline{X}_n \equiv n^{-1} \sum_{i=1}^{n} X_i$ is an unbiased estimator of μ. To provide a measure of the sample mean's precision, an estimate of $\mathrm{Var}[\overline{X}_n]$ also needs to be calculated. If the X_i are independent and identically distributed (i.i.d.) random variables, then the sample variance $S_n^2 \equiv (n-1)^{-1} \sum_{i=1}^{n} (X_i - \overline{X}_n)^2$

is an unbiased estimator of the marginal variance $\sigma_X^2 = \mathrm{Var}[X_1]$, and $\mathrm{Var}[\overline{X}_n]$ can be estimated by S_n^2/n. Unfortunately, the X_i encountered in simulation output are typically correlated. While $\overline{X}_n$ remains an unbiased estimator for μ, the sample variance S_n^2 can be a severely biased estimator for σ_X^2. In fact, if the autocovariance function $R_j \equiv \mathrm{Cov}[X_1, X_{1+j}]$ for $j = 0, \pm 1, \pm 2, \ldots$ is positive, then in many practical applications, one has $\mathrm{E}[S_n^2/n] \ll \mathrm{Var}[\overline{X}_n]$ (Law 2007, Section 4.4), and CI estimation is difficult. However, in such situations, valid CIs for μ can still be obtained based on good estimators of the quantities $\{\sigma_n^2 \equiv n\mathrm{Var}[\overline{X}_n] : n = 1, 2, \ldots\}$ or their limit $\sigma^2 \equiv \lim_{n\to\infty} \sigma_n^2$, which is called the (asymptotic) variance parameter of the process—provided that $\{X_i : i = 1, 2, \ldots\}$ is a process with short-range dependence so that σ^2 is well defined and finite. The above discussion bypasses the effect of the initialization bias for simulations that start in an arbitrary state (Law 2007, Section 9.5.1). This issue is typically addressed by removing batches of observations at the onset of a sample path.

The simulation literature contains many techniques for estimating the quantities σ_n^2 and σ^2, such as: nonoverlapping batch means (NBM) (Fishman 2001; Fishman and Yarberry 1997; Tafazzoli and Wilson 2011); overlapping batch means (OBM) (Meketon and Schmeiser 1984); and standardized time series (STS) (Schruben 1983; Alexopoulos et al. 2007b). Most of these techniques group observations into nonoverlapping or overlapping batches. A method relying on nonoverlapping batches typically divides the data into adjacent disjoint batches of equal size, calculates a corresponding estimator from each batch separately, and then forms an overall variance estimator by averaging the estimators from the various batches. Nonoverlapping batches are used with the following estimators for σ^2: NBM (Steiger et al. 2005), STS area, and STS Cramér–von Mises (CvM) (see Alexopoulos et al. 2007b); folded area and folded CvM (Antonini et al. 2009). The NBM estimator is simply the product of the batch size and the sample variance of the batch means. For STS area and CvM estimators, one forms an STS from each batch, computes an appropriate functional of each STS, and averages these quantities.

This chapter is a step towards the development of effective sequential procedures for computing valid CIs for the steady-state mean of a simulation-generated process using variance estimators based on standardized time series. The algorithms formulated in this chapter are characterized by their simplicity relative to competing methods based on batch means as well as their ability to deliver point estimate CIs for the variance parameter σ^2. Section 3.2 presents a review of the necessary methodology. Section 3.3 details SPSTS, the first of a new class of sequential procedures for steady-state simulation output analysis. Section 3.4 illustrates the performance of SPSTS in test processes that are often used as benchmark problems for evaluating output-analysis procedures. Section 3.5 summarizes our preliminary findings and conclusions.

3.2 Literature Review

The following discussion is necessary to establish basic notation and motivation for the estimators under study herein. Among the assumptions enumerated below, the first property is satisfied by a variety of stationary processes, including strongly mixing processes, associated stationary processes, and regenerative processes (Durrett 2005; Glynn and Iglehart 1990).

3.2.1 Assumptions and Relevant Variance Estimators

Assumptions A

A.1 [Functional Central Limit Theorem (FCLT)] Assume $\sum_{j=-\infty}^{\infty} j^2|R_j| < \infty$ so that we have $\sigma^2 = \lim_{n\to\infty} n\mathrm{Var}[\overline{X}_n] = \sum_{j=-\infty}^{\infty} R_j < \infty$. Further, assume $\sigma^2 > 0$, and that the sequence of random functions

$$Y_n(t) \equiv \frac{\lfloor nt \rfloor (\overline{X}_{\lfloor nt \rfloor} - \mu)}{\sigma\sqrt{n}} \quad \text{for } t \in [0,1] \text{ and } n = 1, 2, \ldots$$

satisfies $Y_n(\cdot) \underset{n\to\infty}{\Longrightarrow} \mathcal{W}(\cdot)$, where: $\lfloor \cdot \rfloor$ is the greatest integer function; $\mathcal{W}(\cdot)$ is a standard Brownian motion process on $[0,1]$; and $\underset{n\to\infty}{\Longrightarrow}$ denotes weak convergence

(as $n \to \infty$) in the Skorohod space $D[0,1]$ of real-valued functions on $[0,1]$ that are right-continuous with left-hand limits.

A.2 The weight function $f(\cdot)$ on $[0,1]$ has a continuous second derivative and is normalized so that $\int_0^1 \int_0^1 f(s)f(t)[\min(s,t) - st]\,ds\,dt = 1$. We define $F(t) \equiv \int_0^t f(s)\,ds$ and $\overline{F}(t) \equiv \int_0^t F(s)\,ds$ for $t \in [0,1]$; and we let $F \equiv F(1)$ and $\overline{F} \equiv \overline{F}(1)$.

Assumption A.1 implies that if $\overline{X}_n$ and $\widehat{\sigma}^2$ are asymptotically independent as $n \to \infty$ and if $\widehat{\sigma}^2 \underset{n \to \infty}{\Longrightarrow} \sigma^2\chi^2_\nu/\nu$, where χ^2_ν denotes a chi-squared distribution with ν degrees of freedom, then

$$\overline{X}_n \pm t_{\nu,1-\alpha/2}\sqrt{\widehat{\sigma}^2/n} \tag{3.2.1}$$

is an asymptotically valid $100(1-\alpha)\%$ CI for μ, where $t_{\nu,\beta}$ is the β quantile of Student's t distribution with ν degrees of freedom depending on the type of the estimator. The absolute precision of the CI (3.2.1) is its half-length, while the relative precision of (3.2.1) is the ratio of its half-length to $\left|\overline{X}_n\right|$.

We define the unknown constants $\gamma_k \equiv 2\sum_{j=1}^{\infty} j^k R_j$, $k = 1, 2, \ldots$. We also use the following notation: (i) $p(n) = O(q(n))$ means that there are positive constants c and n_0 such that $0 \le p(n) \le cq(n)$ for all $n \ge n_0$; and (ii) $p(n) = o(q(n))$ means that $\lim_{n\to\infty} p(n)/q(n) = 0$.

3.2.2 Nonoverlapping Batch Means

The concept of batching has a long history in the simulation output analysis literature. Suppose we form b nonoverlapping batches, each consisting of m observations (so that $n = bm$). Specifically, batch i consists of $\{X_{(i-1)m+j} : j = 1, \ldots, m\}$. Finally, for $i = 1, \ldots, b$ and $k = 1, \ldots, m$, let $\overline{X}_{i,k} \equiv k^{-1}\sum_{\ell=1}^{k} X_{(i-1)m+\ell}$.

One can show that as $m \to \infty$, the nonoverlapping batch means $\overline{X}_{i,m}$ for $i = 1, \ldots, b$ become uncorrelated (Law and Carson 1979) and normally distributed when the process

$\boldsymbol{X}$ obeys an FCLT. Then the NBM estimator for σ^2 is defined by

$$\mathcal{N}(b,m) \equiv \frac{m}{b-1}\sum_{i=1}^{b}(\overline{X}_{i,m} - \overline{X}_n)^2; \tag{3.2.2}$$

that is, m times the sample variance of the batch means. If b is fixed, then $\mathcal{N}(b,m) \underset{m \to \infty}{\Longrightarrow} \sigma^2\chi^2_{b-1}/(b-1)$ and the usual NBM CI estimator for the mean,

$$\overline{X}_n \pm t_{b-1,1-\alpha/2}\sqrt{\frac{\mathcal{N}(b,m)}{n}},$$

is asymptotically valid as $m \to \infty$ (Steiger and Wilson 2001). Under some additional mixing assumptions, one can also show that

$$\mathrm{E}[\mathcal{N}(b,m)] = \sigma^2 - \frac{(b+1)\gamma_1}{bm} + o(\tfrac{1}{m})$$

and

$$\mathrm{Var}[\mathcal{N}(b,m)] = \frac{2\sigma^4(b+1)}{(b-1)^2} + O\Big(\frac{1}{bm^{1/4}}\Big) + o(\tfrac{1}{n});$$

see Alexopoulos and Goldsman (2004). If both m and b tend to ∞, then the last two equations imply weak consistency for $\mathcal{N}(b,m)$ because $\mathrm{MSE}[\mathcal{N}(b,m)] \to 0$.

Several methods based on sequences of batch sizes and batch counts that aim at computing consistent estimators for σ^2 and asymptotically valid CIs for μ have been developed during the last twenty years. The methods of Fishman and Yarberry (1997) take $O(n)$ time and $O(\log_2 n)$ space, and have been implemented in the LABATCH.2 package (available at the Web site `www.or.unc.edu/~gfish/labatch.2.html`). The ASAP3 method (Steiger et al. 2005) has a fixed upper limit on the batch count b but allows $m \to \infty$ so as to deliver approximately valid CIs for μ that satisfy user-specified requirements on absolute or relative precision. This package is available at the Web site `www.ise.ncsu.edu/jwilson`. The Skart procedure (Tafazzoli and Wilson 2011) exploits separate adjustments to the classical batch-means CI to account for the effects on the distribution of the underlying Student's t-statistic arising from skewness and autocorrelation of the batch means. A Visual Basic

implementation of Skart is available online via `www.ise.ncsu.edu/jwilson`. Both ASAP3 and Skart also deliver a point estimator for the steady-state mean that is approximately free of initialization bias.

3.2.2.1 Overview of LABATCH.2

LABATCH.2 is a revision of the LABATCH method of Fishman and Yarberry (1997) that provides an asymptotically valid confidence interval based on the batch means method. Its implementation allows the user to assess the error in the variance estimates as well as the extent to which the sample average is free of initial conditions.

The skeleton of the LABATCH.2 package consists of two rules for updating the batch size and the batch count. Given the initial batching pair (b_1, m_1), the Fixed Number of Batches (FNB) rule fixes $b_j = b_1$ for all j and doubles the batch size $m_{j+1} = 2m_j$ on successive iterations. The Square Root (SQRT) rule defines a batching sequence $\{(b_j, m_j) : j = 1, 2, \ldots\}$ such that $b_{j+1}/b_j \approx \sqrt{2}$ and $m_{j+1}/m_j \approx \sqrt{2}$. More specifically, for given $b_1 > 1$ and m_1, it sets

$$b_2 = \tilde{b}_1 \equiv \lfloor \sqrt{2} b_1 + 0.5 \rfloor,$$

$$m_2 = \tilde{m}_1 \equiv \begin{cases} 3 & \text{if } m_1 = 1 \\ \lfloor \sqrt{2} m_1 + 0.5 \rfloor & \text{if } m_1 > 1, \end{cases}$$

and

$$b_{j+2} = 2b_j, \quad m_{j+2} = 2m_j \quad \text{for } j = 2, 3, \ldots.$$

Choosing (b_1, m_1) from Table 6.3 in Fishman (2001) ensures that $n_j = b_j m_j = 2^{j-1} b_1 m_1$; hence $n_{j+1} = 2n_j$.

Using the above rules, Fishman and Yarberry (1997) constructed two alternative strategies. The LBATCH strategy employs the FNB rule until the batch means pass von Neumann's (1941) test for independence, and then switches to the SQRT rule (without further testing). The more-conservative ABATCH strategy assesses the independence of the batch

means in each iteration. If the test fails to reject independence, then the next iteration uses the SQRT rule; otherwise, it uses the FNB rule.

The LABATCH.2 package has several advantages over other fixed-sample-size methods. First, it is the only package with linear overall time complexity and sublinear space requirements. Second, its dynamic setting allows the user to assess the convergence of the variance of the sample mean as the sample size grows. This assessment is essential to gauge the quality of the CI for the mean. Unfortunately, the LBATCH and ABATCH algorithms are not sequential, in the sense that they cannot deliver CIs for the mean with guaranteed absolute or relative precision. The next two sections review two recent sequential procedures that will be used as "yardsticks" for the proposed methodology.

3.2.2.2 Overview of ASAP3

ASAP3 (Steiger et al. 2005) is a sequential batch means algorithm designed to obtain point and CI estimators for the steady-state mean. ASAP3 proceeds in four steps: (1) the batch size is progressively increased until every other group of four adjacent nonoverlapping batch means pass the Shapiro–Wilk test for quadrivariate normality; (2) then it fits a first-order autoregressive (AR(1)) time series model to the batch means; (3) next ASAP3 computes the terms of an inverse Cornish–Fisher expansion (Stuart and Ord 1994) for the classical NBM Student's t-ratio based on the batch means and the AR(1) parameter estimates; and (4) finally ASAP3 delivers a correlation-adjusted CI for the mean based on latter expansion.

Figure 7 displays a high-level flow chart of ASAP3. In more detail, the first four batches are ignored to reduce the potential effects due to initialization bias and the remaining 252 batch means are organized into adjacent nonoverlapping groups of four consecutive batch means. The set of 32 four-dimensional vectors consisting of every other group is tested for stationary multivariate normality using the Shapiro–Wilk test (Malkovich and Afifi 1973). If this test fails, then the batch size is increased by a factor of $\sqrt{2}$ and the above sequence

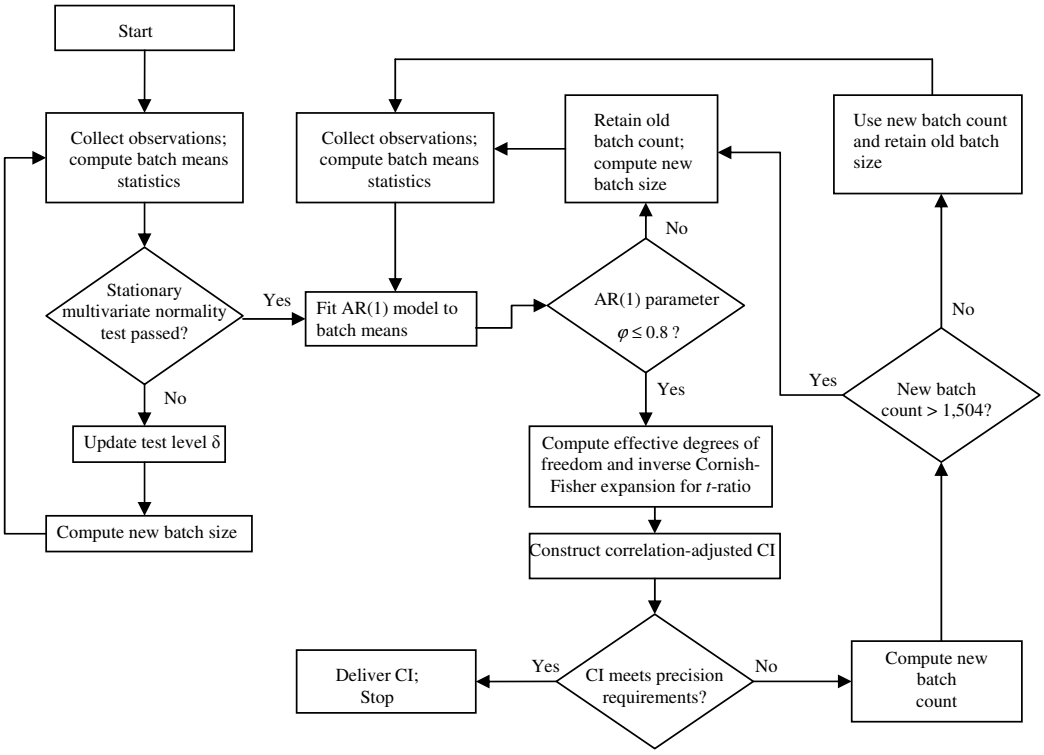


Figure 7: High-level flow chart of ASAP3

of steps is repeated with progressively decreasing significance level (type I error) for the Shapiro–Wilk test (this decrease aims at controlling excessive variability in the final sample size in applications with no precision requirement).

Once the hypothesis of stationary multivariate normality is accepted, ASAP3 fits an AR(1) model to the sequence of 252 batch means and applies a normalizing arc sine transformation to the autoregressive parameter estimator $\widehat{\varphi}$ in order to test the hypothesis that the correlation φ between the batch means is at most 0.8. If the one-sided hypothesis is rejected, the batch size is increased by an appropriate multiplier as detailed in Appendix B of Steiger et al. (2005).

Next, ASAP3 constructs a CI for the mean that has been adjusted to account for correlations between the batch means; this adjustment uses an inverse Cornish–Fisher expansion for the classical Student's t-ratio based on the batch means. If additional observations of

the output process must be obtained before a CI that meets the user-specific precision can be delivered, ASAP3 estimates a new, larger sample size using an approach similar to step **[9]** of algorithm SPSTS in Section 3.3.

In an extensive experimental performance evaluation, Steiger et al. (2005) found that ASAP3 compares favorably with other batch-means procedures (specifically, ABATCH and LBATCH as well as ASAP3's predecessors ASAP and ASAP2) with respect to the following: (a) conformance with the user-specified CI precision and coverage-probability requirements; and (b) mean and variance of the delivered CI half-length.

3.2.2.3 Overview of Skart

Tafazzoli and Wilson (2011) formulated Skart as an automated sequential batch means procedure for constructing a skewness- and autoregression-adjusted confidence interval for the steady-state mean of a simulation output process. Skart is devised to deliver a CI satisfying user-specified requirements concerning coverage probability and absolute or relative precision. Similar to ASAP3, Skart also consists of three main steps.

On the first step, von Neumann's randomness test is sequentially applied to spaced batch means with increasing sizes for each batch and its preceding spacer until the spaced batch means finally pass the test; then the initial spacer is removed to eliminate any warm-up effects.

On step two, from the truncated time series of the original (unbatched) observations, nonspaced batch means are computed using the batch size from the first step; then from estimates of the variance, skewness, and lag-one correlation of the nonspaced batch means, the classical batch means CI for μ is adjusted as follows: (a) the autocorrelation adjustment is based on an AR(1) model for the nonspaced batch means; and (b) the skewness adjustment is based on a Cornish–Fisher expansion for the classical batch means Student's t-statistic.

On the third step, if the precision requirement is satisfied by the CI from the second

step, then the CI is delivered, and Skart terminates; otherwise, the required simulation run length is estimated, the batch count and batch size are suitably increased within certain limits on their growth in one iteration of Skart, and the first two steps are repeated. As in SPSTS, the method for estimating the new batch count and batch size required to satisfy the precision requirement is based on the analogous method developed for ASAP3. The overall structure of Skart is depicted in Figure 8.

Tafazzoli et al. (2011) conducted an extensive experimental performance evaluation of Skart on a wide range of test processes with characteristics that are typical of large-scale practical applications as well as test processes that were deliberately designed to provide an extreme "stress test" of Skart. The experimental results show that Skart compares favorably with other steady-state simulation analysis procedures—namely, its predecessors WASSP (Lada and Wilson 2006), ASAP3, and SBatch, as well as sequential versions of LBATCH and ABATCH, the Law–Carson (1979) procedure, and the spectral procedure of Heidelberger and Welch (1981).

Tafazzoli, Steiger, and Wilson (2011) also developed N-Skart, a nonsequential version of Skart that works with a fixed-size data set. In an experimental performance evaluation involving the same test processes as for Skart, the authors find that N-Skart outperforms LBATCH and ABATCH.

3.2.3 Standardized Time Series

The STS for the sample $\{X_1, \ldots, X_n\}$ is defined by

$$T_n(t) \equiv \frac{\lfloor nt \rfloor (\overline{X}_n - \overline{X}_{\lfloor nt \rfloor})}{\sigma\sqrt{n}} \quad \text{for } t \in [0,1] \text{ and } n = 1, 2, \ldots .$$

Under Assumption FCLT, it can be shown that

$$(\sqrt{n}(\overline{X}_n - \mu), \sigma T_n) \underset{n \to \infty}{\Longrightarrow} (\sigma\mathcal{W}(1), \sigma\mathcal{B}),$$

where $\mathcal{B}(\cdot)$ is a standard Brownian bridge process that is independent of $\mathcal{W}(1)$, so that $T_n(\cdot)$ and $\overline{X}_n$ are asymptotically independent (Schruben 1983).

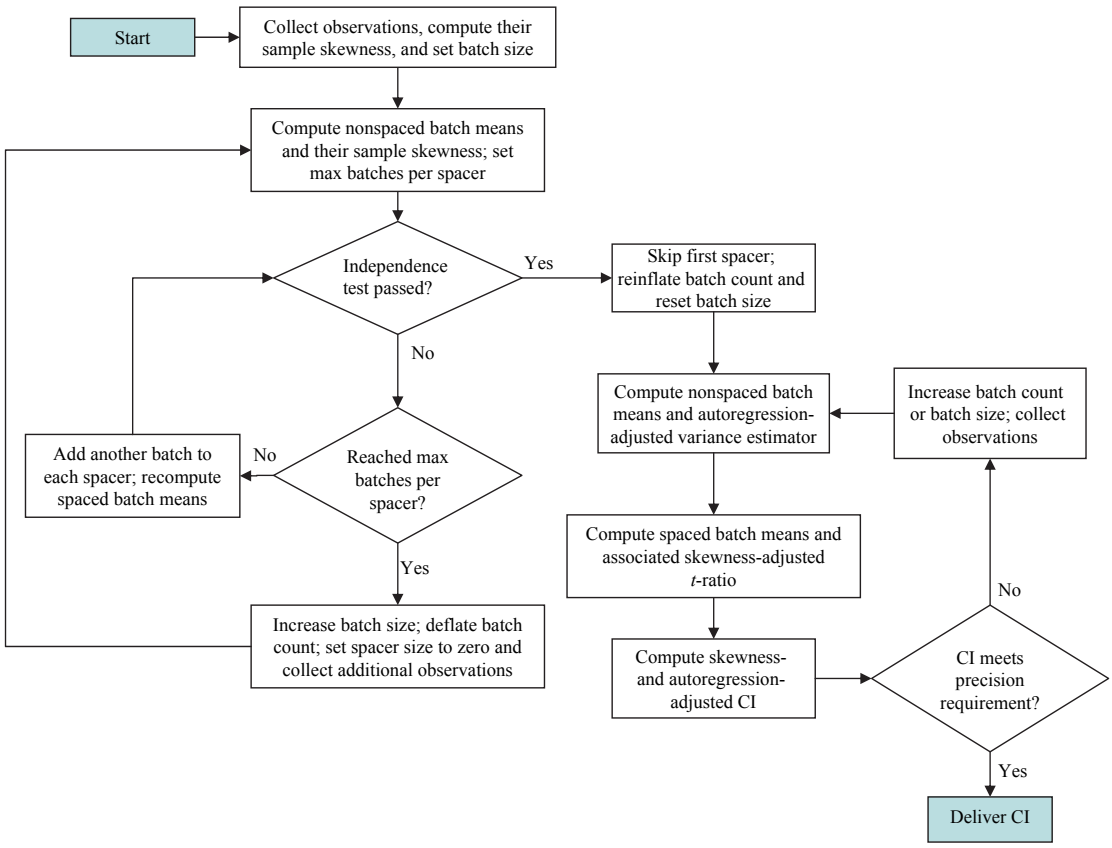


Figure 8: High-level flow chart of Skart

3.2.4 Batched Area Estimators

As in Section 3.2.2, we form b nonoverlapping batches, each consisting of m observations. The STS from batch i is

$$T_{i,m}(t) \equiv \frac{\lfloor mt \rfloor (\overline{X}_{i,m} - \overline{X}_{i,\lfloor mt \rfloor})}{\sigma\sqrt{m}} \quad \text{for } t \in [0,1] \text{ and } i = 1, \ldots, b\,.$$

Under the same mild conditions as above, one has

$$(\sqrt{m}(\overline{X}_{1,m} - \mu), \ldots, \sqrt{m}(\overline{X}_{b,m} - \mu); \sigma T_{1,m}, \ldots, \sigma T_{b,m})$$
$$\Longrightarrow_{m \to \infty} (\sigma\xi_1, \ldots, \sigma\xi_b; \sigma\mathcal{B}_0, \ldots, \sigma\mathcal{B}_{b-1}), \tag{3.2.3}$$

where the ξ_i are i.i.d. standard normal random variables, and $\mathcal{B}_s(\cdot)$ denotes a standard Brownian bridge defined in terms of $\mathcal{W}(\cdot)$ on $[s, s+1]$, for $s \in [0, b-1]$. That is,

$$\mathcal{B}_s(t) \equiv t[\mathcal{W}(s+1) - \mathcal{W}(s)] - [\mathcal{W}(s+t) - \mathcal{W}(s)] \quad \text{for } t \in [0,1] \text{ and } s \in [0, b-1].$$

One can easily show that the Brownian bridges $\mathcal{B}_0(\cdot), \mathcal{B}_1(\cdot), \ldots, \mathcal{B}_{b-1}(\cdot)$ are independent.

We define the *signed areas* associated with the STSs computed from each batch using the weight function $f(\cdot)$ as follows:

$$\begin{aligned} Z_i(f;m) &\equiv \frac{1}{m}\sum_{\ell=1}^{m} f(\tfrac{\ell}{m})\sigma T_{i,m}(\tfrac{\ell}{m}) \\ &= \frac{1}{m^{3/2}}\sum_{\ell=1}^{m} f(\tfrac{\ell}{m})\ell(\overline{X}_{i,m} - \overline{X}_{i,\ell}) \quad \text{for } i = 1,\ldots,b. \end{aligned} \tag{3.2.4}$$

Equation (3.2.3) ensures that

$$\big[Z_1(f;m), Z_2(f;m), \ldots, Z_b(f;m)\big] \underset{m\to\infty}{\Longrightarrow} \sigma\big[\xi_1, \xi_2, \ldots, \xi_b\big] \tag{3.2.5}$$

so that the $\{Z_i(f;m) : i = 1,\ldots,b\}$ become i.i.d. $\sigma N(0,1)$ as $m \to \infty$ (Alexopoulos et al. 2007b, Goldsman et al. 1990).

The *batched area* estimator for σ^2 is the average of the squares of the signed areas:

$$\mathcal{A}(f;b,m) \equiv \frac{1}{b}\sum_{i=1}^{b} Z_i^2(f;m) \underset{m\to\infty}{\Longrightarrow} \sigma^2\chi_b^2/b.$$

Under Assumptions A and some mixing conditions, for fixed b,

$$\mathrm{E}[\mathcal{A}(f;b,m)] = \sigma^2 - \frac{[(F-\overline{F})^2 + \overline{F}^2]\gamma_1}{2m} + o(1/m)$$

and, assuming uniform integrability for the family $\{\mathcal{A}^2(f;b,m) : m = 1,2,\ldots\}$, we have $\lim_{m\to\infty}\mathrm{Var}[\mathcal{A}(f;b,m)] = 2\sigma^4/b$ (Foley and Goldsman 1999; Goldsman et al. 1990). If one chooses weights having $F = \overline{F} = 0$, the resulting area estimator is first-order unbiased since its bias is $o(1/m)$. Damerdji and Goldsman (1995) gave conditions such that the estimator $\mathcal{A}(f;b,m)$ is strongly consistent as both $b, m \to \infty$ in an appropriate fashion.

Popular choices of weight functions are

$$\left.\begin{aligned} f_0(t) &= \sqrt{12} \quad \text{(Schruben 1983)}, \\ f_2(t) &= \sqrt{840}(3t^2 - 3t + 1/2) \quad \text{(Goldsman et al. 1990), and} \\ f_{\cos,j}(t) &= \sqrt{8}\pi j\cos(2\pi jt) \quad j = 1,2,\ldots \quad \text{(Foley and Goldsman 1999)} \end{aligned}\right\} \quad \text{for } t \in [0,1].$$

Schruben's estimator has $\mathrm{E}[\mathcal{A}(f_0; b, m)] = \sigma^2 + 3\gamma_1/m + o(1/m)$, while the remaining estimators are first-order unbiased. Foley and Goldsman (1999) show that the area estimators arising from the weight functions $\{f_{\cos,j}(\cdot) : j = 1, 2, \ldots\}$ are also asymptotically independent. Thus we can average the first k of these area estimators to obtain a first-order unbiased estimator for σ^2 with more degrees of freedom:

$$\overline{\mathcal{A}}_k(f_{\cos}; b, m) \equiv \frac{1}{k}\sum_{j=1}^{k} \mathcal{A}(f_{\cos,j}; b, m) \underset{m \to \infty}{\Longrightarrow} \sigma^2 \chi^2_{bk}/(bk) \quad \text{for } k = 1, 2, \ldots.$$

The following theorem gives detailed expressions for the bias of the batched area estimators $\mathcal{A}(f_{\cos,j}; b, m)$ for $j = 1, 2, \ldots$. This result extends Equation (13) of Aktaran-Kalaycı et al. (2006) which gives the bias of $\mathcal{A}(f_{\cos,1}; b, m)$. The proof can be found in Appendix E.

Theorem 3.2.1. *Suppose that $\{X_1, X_2, \ldots\}$ is covariance stationary with covariance function $\{R_i\}$ satisfying $|R_i| = O(\delta^i)$ for $i = 1, 2, \ldots$, where $\delta \in (0, 1)$. Since $\sum_{i=1}^{\infty} i^k |R_i| < \infty$ for $k = 0, 1, \ldots$, we can define the "inertial" parameters $\gamma_k \equiv 2\sum_{i=1}^{\infty} i^k R_i$. Then for $j \leq j^*$ (where j^* is fixed and $\ll m$), the expected value of the batched area estimator $\mathcal{A}(f_{\cos,j}; b, m)$ is given by*

$$\mathrm{E}[\mathcal{A}(f_{\cos,j}; b, m)] = \sigma^2 - \frac{\pi^2 j^2 (6\gamma_2 - \sigma^2)}{3m^2} + \frac{2\pi^2 j^2 \gamma_3}{3m^3} + \frac{\pi^4 j^4 [\sigma^2 + 10(\gamma_4 - \gamma_2)]}{15m^4} + O(\tfrac{1}{m^5}).$$

Notice that the powers of j permeate through all terms of order ≥ 2. Since often $6\gamma_2 > \sigma^2$, the second-order term of the bias of $\mathcal{A}(f_{\cos,j}; b, m)$ becomes more negative as j increases. For this reason, the proposed sequential procedures rely solely on the area estimators based on the weights $f_{\cos,1}(\cdot)$ and $f_{\cos,2}(\cdot)$.

Example 3.2.1. Consider an $M/M/1$ queueing system with arrival rate ω and traffic intensity $\rho < 1$. Let X_i be the time spent in queue prior to service by entity i, let S_i be the service time for entity i, and let A_i be the time between arrivals $i - 1$ and i. If the delay of the first entity is generated from the c.d.f.

$$P(X_1 \leq x) = 1 - \rho + \rho\left[1 - e^{-\frac{\omega(1-\rho)x}{\rho}}\right], \quad x \geq 0,$$

while the remaining delays are generated using Lindley's recursion

$$X_i = \max\{X_{i-1} + S_{i-1} - A_i, 0\}, \quad i = 2, 3, \ldots,$$

then the process $\{X_1, X_2, \ldots\}$ is stationary with mean $\mu = \rho^2/[\omega(1-\rho)]$ and $\sigma^2 = \rho^3(2 + 5\rho - 4\rho^2 + \rho^3)/[\omega^2(1-\rho^4)]$; see, for example Steiger and Wilson (2001). We consider the case when $\omega = 0.8$ and $\rho = 0.8$; hence $\mu = 8$ and $\sigma^2 = 1976$.

Figure 9 below plots the estimated expected values of $\mathcal{A}(f_{\cos,j}; 1, m)$ for $j = 1, \ldots, 8$ and $m = 2^p$, $p = 7, \ldots, 16$, based on 1000 independent replications. The effect of j towards the size of the negative bias of $\mathcal{A}(f_{\cos,j}; 1, m)$ is apparent.

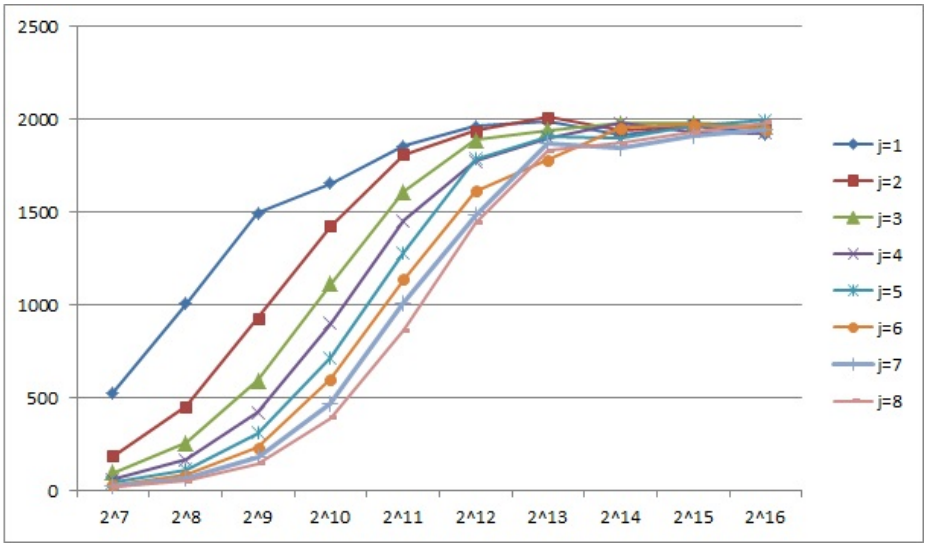


Figure 9: Estimated expected values of area variance estimators based on 8 cosine weight functions for an $M/M/1$ waiting-time process with $\rho = 0.8, \sigma^2 = 1976$, and $b = 1$

3.2.5 Overlapping Batch Means

In various NBM procedures, the von Neumann (1941) randomness test is used to determine a batch size that is sufficiently large to ensure that the resulting batch means are approximately i.i.d. normal with mean μ. Unfortunately in large-scale applications, we often find that the resulting batch means exhibit nonnegligible correlation and skewness (Steiger et al. 2005; Tafazzoli and Wilson 2011). Such correlation usually causes systematic underestimation of $\mathrm{Var}\left[\overline{X}_n\right]$ (that is, negative bias); and this effect coupled with nonnegligible

skewness of the batch means can cause the final CI coverage to fall substantially below its nominal level. ASAP3 and Skart sought to estimate this correlation and compensate for it by modeling the batch means as a stationary AR(1) process for a sufficiently large batch size; moreover Skart also aimed at estimating the skewness of the batch means and then to compensate for it by applying a suitable adjustment to the final CI. This approach proved to effective in a broad diversity of applications.

In the development of SPSTS, we observed similar problems with the von Neumann randomness test and the Shapiro–Wilk normality test when those tests were applied to the signed areas so as to determine a batch size m that is sufficiently large to ensure the $\{Z_i(f;m)\}$ are approximately i.i.d. normal with variance σ^2. In particular, the resulting STS area estimator $\mathcal{A}(f;b,m)$ of the variance parameter σ^2 often exhibits negative second-order bias (for details see Section 3 of Aktaran-Kalaycı et al. 2007 and Theorem 3.2.1); and this effect sometimes causes the final CI coverage to fall below its nominal level. To address this issue, we used variance estimators based on overlapping batch means, which often exhibit lower small-sample bias.

Given a "shift" $d \in \{1,\ldots,m-1\}$, we define the overlapping batches $\{X_{(i-1)d+1},\ldots,X_{(i-1)d+m}\}$ for $i = 1,\ldots,q$, where $q \equiv \lfloor (n-m)/d \rfloor + 1$. Within this context we define $b \equiv n/m$, with the understanding that it does not denote the number of batches. The respective overlapping batch means are

$$\overline{X}^{\mathrm{O}}_{i,m;d} \equiv \frac{1}{m}\sum_{j=1}^{m} X_{(i-1)d+j} \quad \text{for } i = 1,\ldots,q,$$

and the OBM estimator for σ^2 is

$$\mathcal{O}(b,m;d) \equiv \frac{m}{k-1}\sum_{i=1}^{k}(\overline{X}^{\mathrm{O}}_{i,m;d} - \overline{X}_n)^2. \tag{3.2.6}$$

The assignment $d = 1$ yields the estimator of Meketon and Schmeiser (1984). Under the assumptions for the NBM estimator (3.2.2), it can be shown that the OBM variance estimator has about the same mean as NBM, but $\mathrm{Var}[\mathcal{O}(b,m;1)] \approx \frac{2}{3}\mathrm{Var}[\mathcal{N}(b,m)]$.

Welch (1987) noted that the NBM and OBM estimators are related to classical spectral estimators and showed that partial overlapping not only saves computational effort, but retains most benefits of extreme overlapping. For example, $\mathrm{Var}[\mathcal{O}(b,m;m/4)] \approx 0.69\mathrm{Var}[\mathcal{N}(b,m)]$ and $\mathrm{Var}[\mathcal{O}(b,m;m/2)] \approx 0.75\mathrm{Var}[\mathcal{N}(b,m)]$.

When the batch size m is sufficiently large, we can use Theorem 2.1 of Box (1954) to approximate the distribution of $\mathcal{O}(b,m;d)$ by a weighted sum of i.i.d. χ^2_1 random variables. Then the method of Satterthwaite (1941) yields

$$\mathcal{O}(b,m;d) \;\dot\sim\; \mathrm{E}[\mathcal{O}(b,m;d)]\chi^2_\nu/\nu, \;\; \text{where} \;\; \nu = \left\llbracket \frac{2\mathrm{E}^2[\mathcal{O}(b,m;d)]}{\mathrm{Var}[\mathcal{O}(b,m;d)]} \right\rrbracket, \tag{3.2.7}$$

$\llbracket z \rrbracket$ denotes rounding z to the nearest integer, and ν is the "effective" degree of freedom. In particular, for large m we have

$$\nu \approx \begin{cases} \llbracket\, 1.5b \,\rrbracket & \text{for } d = 1 \\ \llbracket\, 1.45b \,\rrbracket & \text{for } d = m/4 \\ \llbracket\, 1.33b \,\rrbracket & \text{for } d = m/2. \end{cases}$$

Using Equation (3.2.7), we obtain the following approximate $100(1-\alpha)\%$ CI for μ:

$$\overline{X}_n \pm t_{\nu,1-\alpha/2}\sqrt{\frac{\mathcal{O}(b,m;d)}{n}}.$$

3.3 SPSTS (Sequential Procedure Based on Standardized Time Series)

In this section, we develop and evaluate SPSTS, the first sequential procedure for estimating the steady-state mean μ based on standardized time series formed by nonoverlapping batches. The methodology is based on Equation (3.2.5) and has two advantages over methods based on batch means: It is relatively simpler and can be used for constructing valid CIs for the variance parameter σ^2 (Alexopoulos et al. 2007a). A high-level flowchart of SPSTS is depicted in Figure 10.

SPSTS starts with $b = 32$ nonoverlapping batches of size $m = 2048$ so that the initial simulation-generated time series $\{X_i\}$ must have length at least $n = bm =$ 65,536.

This starting batching assignment distinguishes SPSTS from NBM-based procedures such as ASAP3 and Skart: The latter methods start with many batches of small size and take advantage of the fact that the nonoverlapping batch means approach passes the normality test faster than the signed estimators $\{Z_i(f;m)\}$ do. However, the AR(1) model applied to the batch means process induces challenges that lead to more-complex algorithmic structures. The substantially large initial batch size used by SPSTS leads to significantly larger average final sample sizes than ASAP3 and Skart under the no-precision requirement; but because of this larger initial batch size, SPSTS avoids the need to adjust for residual correlations between statistics computed from different nonoverlapping batches as required in ASAP3 and Skart. Further, it can make SPSTS data-wasteful when the X_i are sample- or time-averages, e.g., time-averaged queue lengths collected during fixed time windows.

To determine a subsequent batch size m that is sufficiently large so that the $\{Z_i(f;m)\}$ are approximately i.i.d. $\sigma N(0,1)$ (that is, an adequate degree of convergence to the asymptotic result (3.2.5) has been achieved), SPSTS uses sequential versions of the von Neumann randomness test and the Shapiro and Wilk normality test. If the $Z_i(f;m)$ fail iteration ℓ of the von Neumann randomness test ($\ell = 1, 2, \ldots$) at the significance level $\beta\psi(\ell)$, then the batch size m is increased by a factor of about $\sqrt{2}$, the iteration counter r is incremented, and the randomness test is reperformed. We use $\beta = 0.20$ and $\psi(\ell) \equiv \exp\left[-\eta(\ell-1)^\theta\right]$; and the current values of the parameters η and θ are $\eta = 0.184206$ and $\theta = 2$ as specified in Lada and Wilson (2006).

Specifically, steps **[2]**–**[3]** apply the randomness test to each of the signed area sequences $\{Z_i(f_{\text{cos},j};m)\}$ for $j = 1, 2$. Let $m_1(m_2)$ be the batch size when the $\{Z_i(f_{\text{cos},1};m)\}$ (respectively, $\{Z_i(f_{\text{cos},2};m)\}$) pass this test. We proceed with the batch size $m = \max\{m_1, m_2\}$.

Step **[4]** below addresses the simulation *start-up problem* by truncating the first batch of observations. The idea behind this truncation is that once the signed areas pass the randomness test in steps **[2]**–**[3]**, one can assume that the observations beyond the first

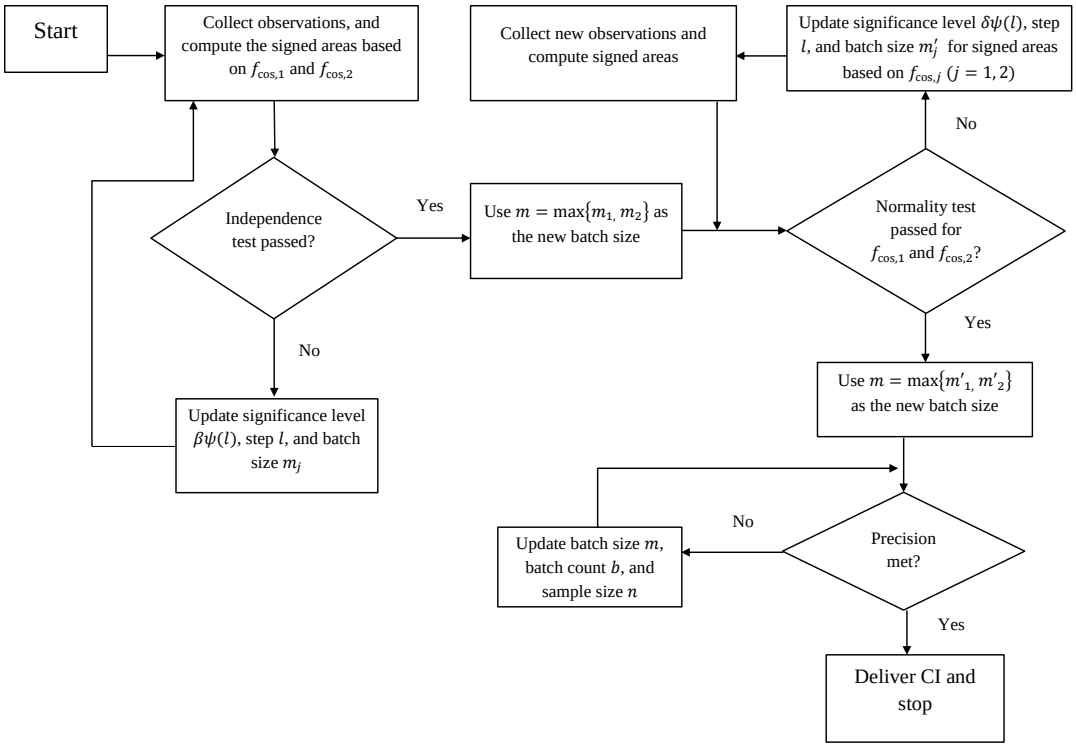


Figure 10: High-level flow chart of SPSTS

batch are free of initialization bias.

Steps **[5]**–**[6]** apply the Shapiro–Wilk normality test. If the $\{Z_i(f; m)\}$ fail iteration ℓ of the normality test (reset $\ell = 1$ when the new test begins) at the significance level $\delta\psi(\ell)$, with $\delta = 0.20$, then the batch size m is increased by a factor of about $\sqrt{2}$, the iteration counter ℓ is incremented, and the normality test is reperformed. As with step **[2]**, let m'_1 (m'_2) be the batch size when the $\{Z_i(f_{\cos,1}; m)\}$ (respectively $\{Z_i(f_{\cos,2}; m)\}$) pass the Shapiro–Wilk test. The algorithm moves on with the batch size $m = \max\{m'_1, m'_2\}$.

The variance estimate in step **[7]** is the conservative maximum of the area estimates based on the cosine weights $f_{\cos,j}$ $(j = 1, 2)$ and the OBM estimate. The effectiveness of this strategy will be assessed experimentally in Section 3.4.4.

Given the user-specified relative precision $r^* > 0$, SPSTS sequentially increases the

total sample size n, the batch size m, and the batch count b in a way similar to ASAP3 and Skart. Specifically, it sets an upper limit on the batch count ($b^* = 64$) and estimates the number of batches b' required to satisfy the precision requirement under the current batch size. The batch count $b = \min\{b', b^*\}$. If b equals b', the batch size remains intact; otherwise, the batch size is increased by a factor equal to the median value of $\{1.05, b'/b, 2\}$. This updating scheme in step **[7]** ensures that the half-length of the final CI for μ does not exceed $r^*\left|\overline{X}_n\right|$ in terms of the final point estimator $\overline{X}_n$ of μ. When $r^* = \infty$ (no-precision case), the algorithm terminates as soon as the signed areas based on the cosine weights $f_{\cos,1}(\cdot)$ and $f_{\cos,2}(\cdot)$ pass both tests for randomness and normality. Finite values of r should be reasonably small, say $r^* \leq 0.15$. A formal algorithmic statement of SPSTS is given below.

Algorithm SPSTS

---

[0] *Initialization:* Set α, $\beta = 0.20$, $\delta = 0.20$, $b^* = 64$, and relative error upper bound r. Define the function $\psi(\ell) \equiv \exp\left[-\eta(\ell-1)^\theta\right]$, $\ell = 1, 2, \ldots$, where $\eta = 0.184206$ and $\theta = 2$.

[1] Generate $b = 32$ batches of size $m = 2048$. Set $j \leftarrow 1$ and $\ell \leftarrow 1$.

[2] **Until** von Neumann's test fails to reject independence using the current weight function $f_j^*(t) \equiv f_{\cos,j}(t), t \in [0,1]$:

- compute the signed areas $\{Z_i(f_j^*; m) : i = 1, \ldots, b\}$ for the weight function $f_j^*(\cdot)$;
- assess the independence of $\{Z_i(f_j^*; m) : i = 1, \ldots, b\}$ using von Neumann's two-sided randomness test at significance level $\beta\psi(\ell)$;
- set $\ell \leftarrow \ell + 1$; generate $\left(\llbracket m\sqrt{2}\,\rrbracket - m\right) b$ additional observations; set $m \leftarrow \llbracket m\sqrt{2}\,\rrbracket$;

[3] Set $m_j \leftarrow m$; reset $m \leftarrow 2048$ and $\ell \leftarrow 1$; update $j \leftarrow j+1$. If $j \leq 2$, then repeat step **[2]**; otherwise, take $m \leftarrow \max\{m_1, m_2\}$ and set $n \leftarrow bm$.

[4] Remove the first batch $\{X_1, \ldots, X_m\}$; reindex the remaining sample and the signed areas as $\{Z_i(f_j^*; m) : i = 1, \ldots, b-1; j = 1, 2\}$; collect a new batch of size m; compute $\{Z_b(f_j^*; m) : j = 1, 2\}$ from the new batch; set $j \leftarrow 1$ and $\ell \leftarrow 1$.

[5] **Until** the Shapiro–Wilk test fails to reject normality using the current weight function $f_j^*(\cdot)$:

- assess the multivariate normality of $\{Z_i(f_j^*; m) : i = 1, \ldots, b\}$ using the Shapiro–Wilk one-sided test at significance level $\delta\psi(\ell)$;
- set $\ell \leftarrow \ell + 1$; generate $\left(\llbracket m\sqrt{2} \rrbracket - m\right) b$ additional observations; set $m \leftarrow \llbracket m\sqrt{2} \rrbracket$;

[6] Set $m_j' \leftarrow m$; reset $m \leftarrow \max\{m_1, m_2\}$ and $\ell \leftarrow 1$; update $j \leftarrow j+1$. If $j \leq 2$, then repeat step **[5]**; otherwise, take $m \leftarrow \max\{m_1', m_2'\}$, and set $n \leftarrow bm$.

[7] **Until** the relative half-length $h(b, m, \alpha)/|\overline{X}_n| \leq r^*$:

[a] Compute the CI midpoint $\overline{X}_n$ and half-length $h(b, m, \alpha)$:

- update the truncated grand mean to reflect the new total sample size $\overline{X}_n \leftarrow \frac{1}{n}\sum_{k=1}^{n} X_k$;
- compute the updated batched area estimates with the current values of b and m:

$$\mathcal{A}(f_j^*; b, m) \equiv \frac{1}{b}\sum_{i=1}^{b} Z_i^2(f_j^*; m) \quad \text{for } j = 1, 2;$$

- compute the updated OBM variance estimate $\mathcal{O}(b, m; m/4)$ using Equation (3.2.6) with shift $d = m/4$ and the current values of n and m;

- set

$$\widehat{V}(b,m) \leftarrow \max\left\{\mathcal{O}(b,m;m/4), \mathcal{A}(f_j^*;b,m) : j=1,2\right\}$$

and compute the half-length

$$h(b,m,\alpha) \equiv t_{\nu,1-\alpha/2}\sqrt{\widehat{V}(b,m)/n}\,,$$

where

$$\nu \approx \begin{cases} [\![\, b \,]\!] & \text{if } \widehat{V}(b,m) = \mathcal{A}(f_j^*;b,m) \\ \lfloor 1.45b \rfloor & \text{if } \widehat{V}(b,m) = \mathcal{O}(b,m;m/4); \end{cases}$$

[b] If $h(b,m,\alpha)/|\overline{X}_n| > r^*$, then

- estimate the number of batches of the current size required to satisfy the precision requirement,

$$b' = \left\lceil b\left\{\frac{h(b,m,\alpha)}{r^*\overline{X}_n}\right\}^2\right\rceil;$$

- update the batch count b, the batch size m, and the total sample size n as follows:

$$b \leftarrow \min\left\{\, b', b^* \,\right\},$$

$$m \leftarrow \begin{cases} m & \text{if } b = b', \\ \lceil m \times \text{mid}\{\, 1.05, \left(b'/b\right), 2 \,\}\rceil & \text{if } b < b', \end{cases}$$

$$n \leftarrow bm\,;$$

- generate the required additional observations.

End If

[8] Deliver the $100(1-\alpha)\%$ CI $\overline{X}_n \;\pm\; h(b,m,\alpha)$.

Remarks: The factor of $\sqrt{2}$ applied to the increase of the batch size in steps **[2]** and **[5]** allows computational savings because accumulators collected in iteration ℓ are used

to compute the $\{Z_i(f_j^*; m)\}$ in iteration $\ell + 2$. Further, for each i and m, the quantities $Z_i(f_1^*; m)$ and $Z_i(f_2^*; m)$ are computed in parallel.

3.4 Simulation Study

This section contains an empirical study designed to test the performance of SPSTS. Specifically, we studied four output processes with increasing statistical challenges: the queue-waiting-time processes in an $M/M/1$ system, an $M/H_2/1$ system and an $M/M/1/$***LIFO*** system, and the AR(1)-to-Pareto (ARTOP) process. The estimates for ASAP3 and Skart are from Table 1 of Tafazzoli et al. (2011). The results for SPSTS and Skart are based on 1000 replications of each procedure, whereas the results for ASAP3 are based on 400 replications.

3.4.1 $M/M/1$ Queue-Waiting-Time Process

We revisit the $M/M/1$ queueing system described in Example 3.2.1. We consider the case where the arrival rate $\omega = 0.9$ and traffic intensity is $\rho = 0.9$, so that $\mu = 9$ and $\sigma^2 = 35{,}901$. Let X_i be the time spent in queue prior to service by entity i. We assume that the system starts empty ($X_1 = 0$).

Table 6 reports results on the following performance characteristics for the SPSTS, Skart, and ASAP3 procedures for CIs with no precision requirement and precisions of 7.5% and 3.75%: empirical coverage, average sample size until procedure termination, average CI half-length, and sample standard deviation of CI half-length. For the specific $M/M/1$ process under study, the CI coverage probabilities for SPSTS usually met or (slightly) exceeded the user-specified nominal levels. Under the no-precision requirement, this was achieved at the cost of substantially larger sample sizes than Skart and ASAP3. (This was anticipated given the discussion in Section 3.3.) However, SPSTS is on equal footing with Skart and ASAP3 with respect to the average CI half-length, i.e., the ratio of the average half-lengths under Skart and SPSTS is close to the square root of the ratio of the average sample sizes required by SPSTS and Skart, respectively. Under the rather stringent

precision requirements of 7.5% and 3.75%, the sample sizes required by SPSTS were about the same as those required for Skart, but somewhat larger than the corresponding sample sizes for ASAP3. Notice that the CIs delivered by SPSTS were noticeably more variable than their counterparts from ASAP3 and Skart.

Table 6: Experimental results for the SPSTS algorithm for an $M/M/1$ system with traffic intensity 0.9

Precision	Performance Measure	Nominal 90% CIs			Nominal 95% CIs		
		SPSTS	Skart	ASAP3	SPSTS	Skart	ASAP3
none	CI coverage (%)	91.5	87.6	87.5	95.2	93.9	91.5
	Avg. sample size	205,210	42,369	31,181	205,210	42,369	31,181
	Avg. CI half-length	0.753	1.767	2.072	0.905	2.298	2.521
	StDev. CI half-length	0.191	0.508	0.590	0.230	1.137	0.731
$\pm 7.5\%$	CI coverage (%)	91.2	91.1	89.5	94.5	95.9	94.0
	Avg. sample size	285,030	302,305	287,568	411,640	431,677	382,958
	Avg. CI half-length	0.600	0.635	0.627	0.603	0.637	0.632
	StDev. CI half-length	0.072	0.037	0.048	0.064	0.035	0.045
$\pm 3.75\%$	CI coverage (%)	89.1	92.0	89.5	94.6	96.0	93.5
	Avg. sample size	1,132,500	1,105,417	969,011	1,552,900	1,586,267	1,341,522
	Avg. CI half-length	0.305	0.321	0.320	0.311	0.321	0.321
	StDev. CI half-length	0.028	0.014	0.020	0.021	0.014	0.020

3.4.2 $M/H_2/1$ Queue-Waiting-Time Process

Consider the waiting-time process in an $M/H_2/1$ queueing system with an empty-and-idle initial condition, an arrival rate of 1.0, and a hyperexponential service-time distribution that is a mixture of two exponential distributions with mixing probabilities $p = (5+\sqrt{15})/10 \approx 0.887$ and $1-p$, and with respective means $1/(2p\tau)$ and $1/[2(1-p)\tau]$, where $\tau = 1.25$. Hence the service times have a mean of 0.8 and a coefficient of variation (standard deviation over mean) of 2.0 (see Appendix 2 of Lada, Steiger, and Wilson 2006). In steady-state operation this system has a server utilization of 80% and a mean queue waiting time of 8.0, so that $\mu = 8.0$ and $\sigma^2 = 24{,}204.8$.

Table 7 reports results of three methods under three precision conditions. For the specific $M/H_2/1$ process under study, the CI coverage probabilities for SPSTS always slightly exceeded the user-specified nominal levels. As anticipated, under the no-precision requirement, the average sample sizes required by SPSTS were substantially larger than the two

competitors, but the CIs produced by SPSTS were tighter. For instance, compare SPSTS and Skart, $\sqrt{179{,}940/30{,}379} = 2.43 < 1.808/0.676 = 2.67$. Under the precision requirements of 7.5% and 3.75%, the sample sizes and the CI coverage probabilities of SPSTS were about the same as those of Skart, but somewhat higher than the corresponding results of ASAP3. Again, the variability of the SPSTS CI half-length is larger than the other two competitors when the relative precision is relatively high (r^* is small).

Table 7: Experimental results for the SPSTS algorithm for an $M/H_2/1$ system with traffic intensity 0.8

Precision	Performance Measure	Nominal 90% CIs			Nominal 95% CIs		
		SPSTS	Skart	ASAP3	SPSTS	Skart	ASAP3
none	CI coverage (%)	93.6	90.0	87.8	96.6	93.0	91.8
	Avg. sample size	179,940	30,379	42,022	179,940	30,379	42,022
	Avg. CI half-length	0.676	1.808	1.614	0.811	2.331	1.950
	StDev. CI half-length	0.161	0.680	0.772	0.194	1.200	0.953
$\pm 7.5\%$	CI coverage (%)	92.7	91.3	90.0	95.7	96.4	94.5
	Avg. sample size	256,700	255,363	228,482	383,030	367,391	309,560
	Avg. CI half-length	0.534	0.566	0.562	0.527	0.566	0.565
	StDev. CI half-length	0.065	0.033	0.045	0.066	0.032	0.017
$\pm 3.75\%$	CI Coverage (%)	91.3	91.8	90.0	96.7	95.9	94.7
	Avg. sample size	965,410	929,527	798,234	1,344,800	1,337,112	1,115,986
	Avg. CI half-length	0.273	0.286	0.287	0.278	0.286	0.288
	StDev. CI half-length	0.022	0.014	0.017	0.018	0.014	0.014

3.4.3 $M/M/1/$*LIFO* Queue-Waiting-Time Process

We now assess the performance of SPSTS based on the queue-waiting-time process in an $M/M/1/$*LIFO* queueing system, with mean interarrival time of 1.0, mean service time 0.8, and LIFO service discipline. Again, X_i is the time spent in queue by entity i prior to service, and we assume that $X_1 = 0$. In steady-state operation, this system has a server utilization of $\tau = 0.8$ and a mean queue waiting time $\mu = 3.2$. The reason for incorporating this process is that in steady state, the observations X_i are highly correlated, with an autocorrelation function that does not exhibit a geometric decay rate, so that substantially large batch sizes hardly reduce the autocorrelation of the batch means (and thereby the signed areas) to negligible levels (see the discussion in Tafazzoli et al. 2011, Section 4.6).

Table 8 summarizes the performance of SPSTS, Skart, and ASAP3. Notice that no results are available for ASAP3 under the 3.75% relative precision requirement. In this case, the CI coverage probabilities for SPSTS always exceeded the user-specified nominal levels. As anticipated, under the no-precision case, the average sample sizes required by SPSTS were substantially larger than for the other two methods, but the respective CI half-lengths were significantly smaller. Under the precision requirement of 7.5%, the sample sizes required by SPSTS were larger than those of Skart and ASAP3. Under the 3.75% relative precision requirement, the sample sizes of SPSTS were about 20% larger than Skart's with nearly the same CI coverage probabilities and average CI half-lengths.

Table 8: Experimental results for the SPSTS algorithm for an $M/M/1/$*LIFO* system with traffic intensity 0.8

Precision	Performance Measure	Nominal 90% CIs			Nominal 95% CIs		
		SPSTS	Skart	ASAP3	SPSTS	Skart	ASAP3
none	CI Coverage (%)	94.2	85.6	87.0	98.5	92.6	92.5
	Avg. sample size	143,185	21,176	53,958	143,185	21,176	53,958
	Avg. CI half-length	0.180	0.514	0.106	0.216	0.699	0.312
	StDev. CI half-length	0.039	0.296	0.106	0.047	0.247	0.008
$\pm 7.50\%$	CI Coverage (%)	93.9	91.6	87.5	98.1	95.9	92.5
	Avg. sample size	144,490	81,441	68,325	153,182	122,391	90,911
	Avg. CI half-length	0.177	0.224	0.219	0.202	0.222	0.226
	StDev. CI half-length	0.035	0.017	0.022	0.031	0.014	0.017
$\pm 3.75\%$	CI Coverage (%)	93.3	91.6		96.5	96.4	
	Avg. sample size	374,815	305,903		516,265	444,852	
	Avg. CI half-length	0.104	0.114		0.106	0.114	
	StDev. CI half-length	0.014	< 0.01		0.012	< 0.01	

3.4.4 AR(1)-to-Pareto (ARTOP) Process

The final testbed for the performance of SPSTS was an AR(1)-to-Pareto process with location parameter $\xi > 0$ and shape parameter $\psi > 0$. This process is generated as follows: one starts with the stationary AR(1) "base process" $\{Z_i : i = 1, 2, \ldots\}$,

$$Z_i = \rho Z_{i-1} + \epsilon_i,$$

where $Z_0 \sim N(0, 1)$ and $\{\epsilon_i : i = 1, 2, \ldots\} \overset{\text{i.i.d.}}{\sim} N(0, \sigma_\epsilon^2)$ is a white-noise process with variance $\sigma_\epsilon^2 = \sigma_Z^2(1 - \rho^2) = 1 - \rho^2$. Then we feed the base process to the standard normal

c.d.f. to obtain a sequence of correlated uniform $U(0,1)$ random variables $\{U_i = \Phi(Z_i) : i = 1, 2, \ldots\}$ where $\Phi(z)$ is the standard normal c.d.f..Then the process $\{U_i : i = 1, 2, \ldots\}$ is treated as input to the inverse of the Pareto c.d.f.

$$F_X(x) \equiv Pr(X \leq x) = \begin{cases} 1 - (\xi/x)^{\psi} & \text{for } x \geq \xi \\ 0 & \text{for } x < \xi \end{cases}$$

to generate the ARTOP process $\{X_i : i = 1, 2, \ldots\}$ by

$$X_i = F_X^{-1}(U_i) = F_X^{-1}(\Phi(Z_i)) = \xi/[1 - \Phi(Z_i)]^{1/\psi} \quad i = 1, 2, \ldots.$$

The mean and variance of the ARTOP process are respectively given by $\mu_X = \mathrm{E}[X_i] = \psi\xi(\psi - 1)^{-1}$ (for $\psi > 1$) and $\sigma_X^2 = \mathrm{Var}[X_i] = \xi^2\psi(\psi - 1)^{-2}(\psi - 2)^{-1}$ (for $\psi > 2$). The parameters of the Pareto distribution were set according to $\psi = 2.1$ and $\xi = 1$, and the lag-one correlation in the base AR(1) process was set to $\rho = 0.995$. This yielded an ARTOP process whose marginal distribution has mean, variance, skewness, and kurtosis, respectively, given by $\mu = 1.9091, \sigma_X^2 = 17.3554$, $\mathrm{E}\{[(X_i - \mu)/\sigma_X]^3\} = \infty$, and $\mathrm{E}\{[(X_i - \mu)/\sigma_X]^4\} = \infty$. The variance parameter $\sigma^2 = 1612.78$.

The most-difficult aspect of this system is that the marginals are highly nonnormal, and their distribution has a very heavy tail. To put SPSTS on equal footing with Skart, we initialized the AR(1) process in state $Z_0 = 3.4$. This yields an initial state $X_0 = F_X^{-1}[\Phi(Z_0)] = 43.5689$, which is 10 standard deviations above the mean μ. On the other hand, ASAP3 was initialized in steady state.

The performance of SPSTS in this challenging instance was similar to the previous two cases. The estimated CI coverage probabilities were remarkably close to those obtained by Skart and significantly better than ASAP3's. As anticipated, under the no-precision requirement, SPSTS required significantly larger sample sizes. Notice that under the 7.5% and 3.75% relative precision requirements, the average CI half-lengths were very close to those delivered by Skart and, most importantly, exhibited comparable variability.

Table 9: Experimental results for the SPSTS algorithm based on an ARTOP Process

Precision	Performance Measure	Nominal 90% CIs			Nominal 95% CIs		
		SPSTS	Skart	ASAP3	SPSTS	Skart	ASAP3
none	CI Coverage (%)	90.3	88.3	85.5	93.0	93.5	90.8
	Avg. sample size	306,590	37,923	114,053	306,590	37,923	114,053
	Avg. CI half-length	0.112	0.640	0.173	0.136	1.030	0.207
	StDev. CI half-length	0.028	0.788	0.099	0.033	2.430	0.120
$\pm 7.50\%$	CI Coverage (%)	89.3	88.3	84.0	92.7	95.7	90.3
	Avg. sample size	322,760	333,666	186,517	362,270	478,926	255,512
	Avg. CI half-length	0.110	0.122	0.127	0.122	0.121	0.131
	StDev. CI half-length	0.021	0.017	0.014	0.018	0.017	0.010
$\pm 3.75\%$	CI Coverage (%)	89.0	91.1	88.8	95.3	96.6	91.0
	Avg. sample size	1,035,300	1,098,130	734,312	1,434,100	1,588,612	1,044,259
	Avg. CI half-length	0.062	0.064	0.067	0.063	0.064	0.067
	StDev. CI half-length	0.008	< 0.01	< 0.01	0.007	< 0.01	< 0.01

Remarks: The ARTOP process provides an avenue for assessing the effectiveness of step **[7]** of Algorithm SPSTS, where $\widehat{V}(b, m)$ is the maximum of the three variance estimates. Table 10 below contains experimental results based on 1000 replications of nominal 90% CI estimators of μ that are generated by three versions of step **[7]** in SPSTS using the following variance estimators:

- the maximum estimator

$$\widehat{V}(b, m) = \max\left\{\mathcal{O}(b, m; m/4), \mathcal{A}(f_{\cos,j}; b, m) : j = 1, 2\right\} \tag{3.4.1}$$

  as in step **[7]** of the original SPSTS procedure;

- $\widehat{V}(b, m) = \mathcal{A}(f_{\cos,1}; b, m)$ in the first alternative version of step **[7]**; and

- $\widehat{V}(b, m) = \mathcal{O}(b, m; m/4)$ in the second alternative version of step **[7]**.

These results show that the CIs based on a single variance estimator fail to achieve the nominal CI coverage. We wish to point out that we experienced similar behaviors with the previous three systems.

3.5 Discussion and Conclusions

This chapter introduced SPSTS, an automated sequential procedure for computing point and CI estimators for the steady-state mean of a simulation output process. SPSTS is based

Table 10: Performance of nominal 90% CIs for μ in an ARTOP process using three alternatives for $\widehat{V}(b,m)$ in step **[7]** of Algorithm SPSTS

Precision	Performance Measure	$\widehat{V}(b,m) =$ Eq. (3.4.1)	$\mathcal{A}(f_{\cos,1};b,m)$	$\mathcal{O}(b,m;m/4)$
none	CI coverage (%)	89.3	84.4	86.5
	Avg. sample size	306,590	306,590	306,590
	Avg. CI half-length	0.114	0.1	0.104
	StDev. CI half-length	0.028	0.027	0.026
3.75%	CI coverage (%)	89.2	84.8	86.6
	Avg. sample size	1,040,300	798,370	816,470
	Avg. CI half-length	0.062	0.061	0.063
	StDev. CI half-length	0.008	0.009	0.008

on STS area estimators; and in an empirical evaluation involving three queueing systems and the ARTOP process, we found that SPSTS performed reasonably well compared with two state-of-the-art sequential procedures based on nonoverlapping batch means in terms of CI coverage, average required sample size, and the first two moments of the CI's half-length. The next chapter develops an algorithm based on STS area variance estimators computed from overlapping batches.

CHAPTER IV

SEQUENTIAL PROCEDURES WITH OVERLAPPING ESTIMATORS FOR ESTIMATING THE STEADY-STATE MEAN

4.1 Introduction

Although the skeleton of Algorithm SPSTS is based on estimators constructed from nonoverlapping standardized time series, step **[7]** used the variance estimator constructed from overlapping batch means to resolve issues related to residual autocorrelation and marginal skewness of the signed areas. The OBM methodology was introduced by Meketon and Schmeiser (1984). Goldsman and Meketon (1984) and Song and Schmeiser (1993, 1995) studied the first two moments of the OBM estimator, and Darmerdji (1994) established the asymptotic consistency of the OBM estimator under properly increasing sequences of batch sizes.

The first attempt towards the determination of a batch size that minimizes the mean squared error of the OBM estimator was made in an unpublished technical report by Pedrosa and Schmeiser (1994); unfortunately, the algorithm is not sequential since it does not compute a CI for μ that achives user-specified coverage probability and precision. Loosely speaking, SPSTS could be considered as the first OBM-based sequential procedure (when $\widehat{V}(b,m) = \mathcal{O}(b,m;m/4)$ in step **[7]**), but the experimental results in Table 10 illustrate that the sole use of OBM variance estimators can result in CIs for μ that fail to achieve the nominal coverage probability.

The foundation for overlapping variance estimators based on STS was laid by Alexopoulos et al. (2007a, 2007b), who obtained detailed expressions for the first two moments of these estimators, consistency in mean square as both the batch size m and the ratio

$b = n/m$ of the sample size to the batch size tend to infinity, and the asymptotic distributions of these estimators as m goes to infinity while b remains constant. Moreover, they showed that the latter asymptotic distributions can be approximated remarkably well by scaled chi-square distributions. Compared with their counterparts computed from batch means or nonoverlapping STS, the overlapping estimators asymptotically achieve substantially reduced variance while maintaining the same bias as the sample size increases.

In this chapter, we propose the first sequential algorithm for estimating the steady-state mean based solely on overlapping variance estimators. This algorithm maintains the structure of SPSTS, but the use of overlapping STS area variance estimators in step **[7]** aims at reducing the sample size requirements under stringent precision constraints for the half-length of the resulting CI. The remainder of this chapter proceeds as follows. Section 4.2 reviews properties of overlapping STS area estimators. Section 4.3 develops the proposed sequential procedure SPOSTS. Section 4.4 evaluates the performance of this sequential procedure against SPSTS and the NBM-based methods Skart and ASAP3. Section 4.5 summarizes our preliminary findings.

4.2 Estimation Based on Overlapping Batches

In Section 3.2.5 we discussed variance estimators based on sample means of overlapping batches of observations and mentioned their lower variance than the NBM estimator. As in Alexopoulos et al. (2007a, 2007b), variance estimators can also be obtained from STSs based on overlapping batches. We will briefly recall the OBM method and then will shift focus towards overlapping STS area estimators. All estimators are based on the shift $d = 1$, the only shift studied by Alexopoulos et al. (2007a, 2007b).

4.2.1 The OBM Variance Estimator

Given the shift $d = 1$, the overlapping batches are $\{X_i, \ldots, X_{i+m-1}\}$ for $i = 1, \ldots, n - m + 1$, the respective overlapping batch means are

$$\overline{X}^{\mathrm{O}}_{i,m;1} \equiv \frac{1}{m} \sum_{j=1}^{m} X_{i+j-1} \quad \text{for } i = 1, \ldots, n - m + 1,$$

and the OBM estimator for σ^2 is

$$\mathcal{O}(b, m; 1) \equiv \frac{m}{n - m} \sum_{i=1}^{n-m+1} (\overline{X}^{\mathrm{O}}_{i,m;1} - \overline{X}_n)^2. \tag{4.2.1}$$

4.2.2 Overlapping STS Area Estimators

The STS computed from overlapping batch i (namely, $\{X_i, X_{i+1}, \ldots, X_{i+m-1}\}$) is

$$T^{\mathrm{O}}_{i,m}(t) \equiv \frac{\lfloor mt \rfloor (\overline{X}^{\mathrm{O}}_{i,m;1} - \overline{X}^{\mathrm{O}}_{i,\lfloor mt \rfloor;1})}{\sigma\sqrt{m}} \quad \text{for } t \in [0, 1] \text{ and } i = 1, \ldots, n - m + 1;$$

the respective signed areas are

$$\begin{aligned} Z^{\mathrm{O}}_i(f; m) &\equiv \frac{1}{m} \sum_{\ell=1}^{m} f(\tfrac{\ell}{m}) \sigma T^{\mathrm{O}}_{i,m}(\tfrac{\ell}{m}) \\ &= \frac{1}{m^{3/2}} \sum_{\ell=1}^{m} f(\tfrac{\ell}{m}) \ell \left(\overline{X}^{\mathrm{O}}_{i,m;1} - \overline{X}^{\mathrm{O}}_{i,\ell;1} \right) \quad \text{for } i = 1, \ldots, n - m + 1; \end{aligned}$$

and the overlapping area estimator for σ^2 is the average

$$\mathcal{A}^{\mathrm{O}}(f; b, m) \equiv \frac{1}{n - m + 1} \sum_{i=1}^{n-m+1} \left[Z^{\mathrm{O}}_i(f; m) \right]^2. \tag{4.2.2}$$

Under Assumptions A in Chapter 3, Alexopoulos et al. (2007b) show that for fixed b,

$$\mathcal{A}^{\mathrm{O}}(f; b, m) \underset{m \to \infty}{\Longrightarrow} \mathcal{A}^{\mathrm{O}}(f; b) \equiv \frac{1}{b - 1} \int_0^{b-1} \left[\sigma \int_0^1 f(u) \mathcal{B}_s(u)\, du \right]^2 ds. \tag{4.2.3}$$

where $\mathcal{B}_s(\cdot)$ denotes a standard Brownian bridge defined in terms of $\mathcal{W}(\cdot)$ on $[s, s + 1]$, for $s \in [0, b - 1]$. Although $\mathrm{E}[\mathcal{A}^{\mathrm{O}}(f; b, m)] = \mathrm{E}[\mathcal{A}(f; b, m)]$, overlapping yields estimators with reduced variance compared with their nonoverlapping counterparts; see Table 11 for details. For instance, for large b, $\mathrm{Var}[\mathcal{A}^{\mathrm{O}}(f_0; b, m)]/\mathrm{Var}[\mathcal{A}(f_0; b, m)] \approx 12/35$, with

similar impressive savings for the overlapping area estimators based on the weights f_2 and $f_{\cos,j}$. Additional benefits can be obtained by averaging the estimators $\mathcal{A}^{\mathrm{O}}(f_{\cos,j}; b, m)$ over $j = 1, \ldots, k$, say, despite the dependencies due to the overlapping batches. The averaged estimators are denoted in Table 11 by $\overline{\mathcal{A}}_k^{\mathrm{O}}(f_{\cos}; b, m) \equiv \frac{1}{k}\sum_{j=1}^{k} \mathcal{A}^{\mathrm{O}}(f_{\cos,j}; b, m)$. This asymptotic reduction in variance should result in less variable CI half-lengths, provided the sample size is sufficiently large.

Despite the intricate calculations, each overlapping area estimator (as well as the OBM estimator) can be computed in $O(n)$ time. The details can be found in the online supplement of Alexopoulos et al. (2007a).

Table 11: Asymptotic bias and variance of STS area variance estimators as $m \to \infty$

Nonoverlapping Estimators			Overlapping Estimators			
Form of Estimator	$\frac{m}{\gamma}$ Bias	$\frac{b}{\sigma^4}$ Var	Form of Estimator	$\frac{m}{\gamma}$ Bias	Var/σ^4	$\approx \frac{b}{\sigma^4}$ Var
$\mathcal{A}(f_0; b, m)$	3	2	$\mathcal{A}^{\mathrm{O}}(f_0; b, m)$	3	$\frac{24b-31}{35(b-1)^2}$	0.686
$\mathcal{A}(f_2; b, m)$	$o(1)$	2	$\mathcal{A}^{\mathrm{O}}(f_2; b, m)$	$o(1)$	$\frac{3514b-4359}{4290(b-1)^2}$	0.819
$\mathcal{A}(f_{\cos,j}; b, m)$	$o(1)$	2	$\mathcal{A}^{\mathrm{O}}(f_{\cos,j}; b, m)$	$o(1)$	$\frac{8\pi^2 j^2+15}{12\pi^2 j^2}$	
$\mathcal{A}(f_{\cos,1}; b, m)$	$o(1)$	2	$\mathcal{A}^{\mathrm{O}}(f_{\cos,1}; b, m)$	$o(1)$	$\frac{8\pi^2+15}{12\pi^2}$	
$\mathcal{A}(f_{\cos,2}; b, m)$	$o(1)$	2	$\mathcal{A}^{\mathrm{O}}(f_{\cos,2}; b, m)$	$o(1)$	$\frac{32\pi^2+15}{48\pi^2}$	
$\overline{\mathcal{A}}_2(f_{\cos}; b, m)$	$o(1)$	1	$\overline{\mathcal{A}}_2^{\mathrm{O}}(f_{\cos}; b, m)$	$o(1)$	$\frac{(384\pi^2+1090)b-(480\pi^2+1455)}{1152\pi^2(b-1)^2}$	0.429
$\mathcal{N}(b, m)$	$\frac{b+1}{b}$	$\frac{2b}{b-1}$	$\mathcal{O}(b, m; 1)$	$\frac{b^2+1}{b(b-1)}$	$\frac{4b^3-11b^2+4b+6}{3(b-1)^4}$	1.333

Alexopoulos et al. (2007a, Sections 4.2–4.3) also obtained approximations for the distribution of overlapping estimators and used those approximations to obtain approximate CIs for μ.

In particular, when m is sufficiently large, it follows from Theorem 2.1 of Box (1954) that the distribution of $\mathcal{A}^{\mathrm{O}}(f; b, m)$ can be approximated by a weighted sum of i.i.d. χ_1^2 random variables. Then the technique of Satterthwaite (1941) yields

$$\mathcal{A}^{\mathrm{O}}(f; b, m) \;\dot{\sim}\; \mathrm{E}[\mathcal{A}^{\mathrm{O}}(f; b, m)]\chi_\nu^2/\nu, \;\; \text{where} \;\; \nu = \left[\!\!\left[\frac{2\mathrm{E}^2[\mathcal{A}^{\mathrm{O}}(f; b, m)]}{\mathrm{Var}[\mathcal{A}^{\mathrm{O}}(f; b, m)]} \right]\!\!\right] \tag{4.2.4}$$

is the effective degrees of freedom. Using Equation (4.2.4), one has the following approximate $100(1-\alpha)\%$ CI for μ:

$$\overline{X}_n \pm t_{\nu,1-\alpha/2}\sqrt{\frac{\mathcal{A}^{\mathrm{O}}(f;b,m)}{n}}. \tag{4.2.5}$$

As we discussed in Section 3.3, Equations (4.2.4)–(4.2.5) are applicable to the OBM estimator (4.2.1) as well. In summary, we have

$$\nu \approx \begin{cases} \left\llbracket \dfrac{24\pi^2 j^2 b}{8\pi^2 j^2 + 15} \right\rrbracket & \text{for } \mathcal{A}^{\mathrm{O}}(f_{\cos,j};b,m),\ j=1,2 \\ \\ \llbracket 1.5b \rrbracket & \text{for } \mathcal{O}(b,m;1). \end{cases} \tag{4.2.6}$$

4.3 SPOSTS (Sequential Procedure Based on Overlapping Standardized Time Series)

In this section, we develop and evaluate SPOSTS, the first sequential procedure for estimating the steady-state mean μ based on overlapping estimators. While this algorithm retains the advantage of SPSTS over sequential methods based on batch means, the lower asymptotic variance of the overlapping STS area estimators should result in more-accurate estimates of the total sample size required to satisfy the precision requirement on each iteration of the procedure; and this in turn should reduce the overall number of iterations required to deliver the final CI estimator of μ with the user's specified level of precision. For this reason we expect SPOSTS to require smaller average sample sizes, especially in the case of a stringent precision requirement (small r^*).

SPOSTS also starts with $b = 32$ nonoverlapping batches of size $m = 2048$ so that the initial simulation-generated time series $\{X_i\}$ must have length not less than $n = bm =$ 65,536. This initial batch size used by SPOSTS leads to significantly larger average final sample sizes than ASAP3 and Skart.

Since the signed areas computed from overlapping batches are dependent random variables, we instead use the b nonoverlapping signed areas $Z_i(b,m)$ in Equation (3.2.4) to

resolve the start-up problem and to assess the normality (Shapiro–Wilk test) of the signed areas based on the $n - m + 1$ overlapping batches.

Algorithm SPOSTS proceeds in a fashion similar to Procedure SPSTS until step **[7]**, which involves the calculation of the variance estimates and the approximate CI for the mean μ. In particular, steps **[2]–[3]** apply von Neumann's (1941) randomness test to the nonoverlapping signed areas $Z_i(f_{\cos,j}; b, m)$ with increasing batch sizes and decreasing significance levels until both sequences pass the test; and then it sets m equal to the largest of the two batch sizes. The idea is that this batch size should be sufficient for the sequences of the $n - m + 1$ *overlapping* signed areas to have a multivariate normal distribution.

Step **[7]** proceeds analogously to the respective step in Algorithm SPSTS. Step **[8]** delivers the CI for μ. We proceed with a formal statement of the algorithm.

Algorithm SPOSTS

---

[0] *Initialization:* Set $\beta = 0.20$, $\delta = 0.20$, $b^* = 64$, the user-specified relative error, and the user-specified CI confidence coefficient $\alpha \in (0, 1)$. Define the function $\psi(\ell) \equiv \exp\big[-\eta(\ell - 1)^\theta\big]$, $\ell = 1, 2, \ldots$, where $\eta = 0.184206$ (the significance level diminishes to 0 in five steps) and $\theta = 2$.

[1] Generate $b = 32$ batches of size $m = 2048$. Set $j \leftarrow 1$ and $\ell \leftarrow 1$.

[2] **Until** von Neumann's test fails to reject independence using the current weight function $f_j^*(t) \equiv f_{\cos,j}(t)$, $t \in [0, 1]$:

- compute the nonoverlapping signed areas $\{Z_i(f_j^*; m) : i = 1, \ldots, b\}$ for the weight function $f_j^*(\cdot)$;
- assess the independence of $\{Z_i(f_j^*; m) : i = 1, \ldots, b\}$ using von Neumann's two-sided randomness test at significance level $\beta\psi(\ell)$;
- set $\ell \leftarrow \ell + 1$, generate $\big(\llbracket m\sqrt{2}\,\rrbracket - m\big)\, b$ additional observations; set $m \leftarrow \llbracket m\sqrt{2}\,\rrbracket$;

[3] Set $m_j \leftarrow m$; reset $m \leftarrow 2048$ and $\ell \leftarrow 1$; update $j \leftarrow j+1$. If $j \leq 2$, then repeat step **[2]**; otherwise, take $m \leftarrow \max\{m_1, m_2\}$ and set $n \leftarrow bm$.

[4] Remove the first batch $\{X_1, \ldots, X_m\}$; reindex the remaining sample and the signed areas as $\{Z_i(f_j^*; m) : i = 1, \ldots, b-1; j = 1, 2\}$; collect a new batch of size m; compute $\{Z_b(f_j^*; m) : j = 1, 2\}$ from the new batch; set $j \leftarrow 1$ and $\ell \leftarrow 1$.

[5] **Until** the Shapiro–Wilk test fails to reject normality using the current weight function $f_j^*(\cdot)$:

- assess the multivariate normality of $\{Z_i(f_j^*; m) : i = 1, \ldots, b\}$ using the Shapiro–Wilk one-sided test at significance level $\delta\psi(\ell)$;
- set $\ell \leftarrow \ell + 1$, generate $\left(\llbracket m\sqrt{2} \rrbracket - m\right) b$ additional observations; set $m \leftarrow \llbracket m\sqrt{2} \rrbracket$;

[6] Set $m_j' \leftarrow m$; reset $m \leftarrow \max\{m_1, m_2\}$ and $\ell \leftarrow 1$; update $j \leftarrow j+1$. If $j \leq 2$, then repeat step **[5]**; otherwise, take $m \leftarrow \max\{m_1', m_2'\}$, and set $n \leftarrow bm$.

[7] **Until** the relative half-length $h(b, m, \alpha)/|\overline{X}_n| \leq r^*$:

[a] Compute the CI midpoint $\overline{X}_n$ and half-length $h(b, m, \alpha)$:

- update the truncated grand mean, $\overline{X}_n \leftarrow \frac{1}{n}\sum_{k=1}^{n} X_k$;
- compute the updated overlapping area estimates $\mathcal{A}^{\mathrm{O}}(f_j^*; b, m)$ for $j = 1, 2$ using Equation (4.2.2);
- compute the updated OBM variance estimate $\mathcal{O}(m, n; 1)$ using Equation (4.2.1) with shift $d = 1$;
- set

$$\widehat{V}(b, m) \leftarrow \max\left\{\mathcal{O}(b, m; 1), \mathcal{A}^{\mathrm{O}}(f_j^*; b, m) : j = 1, 2\right\},$$

$$\nu \leftarrow \begin{cases} \left[\!\left[\dfrac{24\pi^2 j^2 b}{8\pi^2 j^2 + 15} \right]\!\right] & \text{if } \widehat{V}(b,m) = \mathcal{A}^{\mathrm{O}}(f_j^*; b, m),\ j = 1, 2 \\ [\![\, 1.5b \,]\!] & \text{if } \widehat{V}(b,m) = \mathcal{O}(b, m; 1) \end{cases}$$

and compute the half-length

$$h(b, m, \alpha) \equiv t_{\nu, 1-\alpha/2} \sqrt{\widehat{V}(b,m)/n};$$

[b] If $h(b, m, \alpha)/|\overline{X}_n| > r^*$, then

- estimate the sample-to-batch size ratio required to satisfy the precision requirement

$$b' = \left\lceil b \left\{ \frac{h(b, m, \alpha)}{r^* \overline{X}_n} \right\}^2 \right\rceil ;$$

- update the sample-to-batch size ratio b, the batch size m, and the total sample size n as follows,

$$b \leftarrow \min\left\{\, b', b^* \,\right\},$$

$$m \leftarrow \begin{cases} m & \text{if } b = b', \\ \left\lceil m \times \text{mid}\left\{\, 1.05, \left(b'/b\right), 2 \,\right\} \right\rceil & \text{if } b < b', \end{cases}$$

$$n \leftarrow bm\,.$$

- generate the required additional observations.

End If

[8] Deliver the $100(1-\alpha)\%$ CI $\overline{X}_n \ \pm \ h(b, m, \alpha)$.

4.4 Simulation Study

This section uses an empirical study to assess the effectiveness of Algorithm SPOSTS. As in Section 3.4, we studied four output processes with increasing statistical challenges: the waiting-time processes in an $M/M/1$ system, an $M/H_2/1$ system, an $M/M/1/$***LIFO*** system, and the AR(1)-to-Pareto (ARTOP) process. The estimates for ASAP3 and Skart are from Table 1 of Tafazzoli et al. (2011). The results for SPOSTS, SPSTS and Skart are based on 1000 replications of each procedure, whereas the results for ASAP3 are based on 400 replications.

4.4.1 $M/M/1$ Queue-Waiting-Time Process

Consider an $M/M/1$ queueing system with an interarrival rate of 0.9 and traffic intensity 0.9, and assume that the system starts empty and idle.

Table 12 reports results on the following performance characteristics for CIs for the steady-state mean delay in queue based on the SPOSTS, SPSTS, Skart, and ASAP3 procedures under the no-precision requirement and precisions of 7.5% and 3.75%: empirical coverage, average final sample size, average CI half-length, and sample standard deviation of CI half-length. For the specific $M/M/1$ process under study, the CI coverage probabilities for SPOSTS usually met the user-specified nominal levels. Although under the no-precision requirement, this was achieved at the cost of substantially larger sample sizes than Skart and ASAP3, SPOSTS (like SPSTS) is on equal footing with Skart and ASAP3 with regard to the average CI half-length; this indicates the efficiency of our proposed method. Under the rather stringent precision requirements of 7.5% and 3.75%, the sample sizes required by SPOSTS were on the average 8%–10% smaller than those required for SPSTS, and about 7%–12% smaller than those required by Skart.

4.4.2 $M/H_2/1$ Queue-Waiting-Time Process

Consider again the waiting-time process in an $M/H_2/1$ queueing system with an empty-and-idle initial condition, an arrival rate of 1.0, and a hyperexponential service-time

Table 12: Experimental results for the SPOSTS algorithm for an $M/M/1$ system with traffic intensity 0.9

Precision	Performance Measure	Nominal 90% CIs				Nominal 95% CIs			
		SPOSTS	SPSTS	Skart	ASAP3	SPOSTS	SPSTS	Skart	ASAP3
none	CI coverage (%)	89.4	91.5	87.6	87.5	94.8	95.2	93.9	91.5
	Avg. sample size	205,210	205,210	42,369	31,181	205,210	205,210	42,369	31,181
	Avg. CI half-length	0.740	0.753	1.767	2.072	0.886	0.905	2.298	2.521
	StDev. CI half-length	0.197	0.191	0.508	0.590	0.236	0.230	1.137	0.731
$\pm 7.5\%$	CI coverage (%)	88.8	91.2	91.1	89.5	94.5	94.5	95.9	94.0
	Avg. sample size	265,250	286,030	302,305	287,568	376,020	411,640	431,677	382,958
	Avg. CI half-length	0.600	0.600	0.635	0.627	0.611	0.603	0.637	0.632
	StDev. CI half-length	0.070	0.072	0.037	0.048	0.062	0.064	0.035	0.045
$\pm 3.75\%$	CI coverage (%)	89.9	89.1	92.0	89.5	95.3	94.6	96.0	93.5
	Avg. sample size	1,026,300	1,132,500	1,105,417	969,011	1,407,400	1,552,900	1,586,267	1,341,522
	Avg. CI half-length	0.315	0.305	0.321	0.320	0.322	0.311	0.321	0.321
	StDev. CI half-length	0.027	0.028	0.014	0.020	0.020	0.021	0.014	0.020

distribution that is a mixture of two exponential distributions with mixing probabilities $p = (5 + \sqrt{15})/10 \approx 0.887$ and $1 - p$, and respective means $1/(2p\tau)$ and $1/[2(1 - p)\tau]$, where $\tau = 1.25$. In steady-state operation this system has a server utilization of 80% and a mean queue waiting time of 8.0.

Table 13 reports results of the four methods under the three precision conditions. For the specific $M/H_2/1$ process under study, the CI coverage probabilities for SPOSTS always met or slightly exceeded the user-specified nominal levels. Under the no-precision requirement, the average sample sizes required by SPOSTS were the same as SPSTS and were substantially larger than for Skart and ASAP3, but the CIs produced by SPOSTS were noticeably narrower. Under the precision requirements of 7.5% and 3.75%, the CI coverage probabilities of SPOSTS were about the same as those of SPSTS and Skart, but required smaller average sample sizes. The sample sizes and the CI coverage probabilities were somewhat higher than the corresponding results of ASAP3.

4.4.3 $M/M/1/LIFO$ Queue-Waiting-Time Process

We now evaluate the performance of SPOSTS based on the queue-waiting-time process in an $M/M/1/LIFO$ queueing system with an empty-and-idle initial condition. Customers in the system are served in LIFO order, with a mean interarrival time of 1.0, and a mean service time of 0.8. In steady-state, this system has a server utilization of $\tau = 0.8$ and a mean queue waiting time $\mu = 3.2$.

Table 13: Experimental results for the SPOSTS algorithm for an $M/H_2/1$ system with traffic intensity 0.8

Precision	Performance Measure	Nominal 90% CIs				Nominal 95% CIs			
		SPOSTS	SPSTS	Skart	ASAP3	SPOSTS	SPSTS	Skart	ASAP3
none	CI coverage (%)	93.5	93.6	90.0	87.8	96.0	96.6	93.0	91.8
	Avg. sample size	179,940	179,940	30,379	42,022	179,940	179,940	30,379	42,022
	Avg. CI half-length	0.663	0.676	1.808	1.614	0.793	0.811	2.331	1.950
	StDev. CI half-length	0.166	0.161	0.680	0.772	0.199	0.194	1.200	0.953
$\pm 7.5\%$	CI coverage (%)	92.2	92.7	91.3	90.0	95.4	95.7	96.4	94.5
	Avg. sample size	240,110	256,700	255,363	228,482	342,770	384,030	367,391	309,560
	Avg. CI half-length	0.538	0.534	0.566	0.562	0.539	0.527	0.566	0.565
	StDev. CI half-length	0.063	0.065	0.033	0.045	0.060	0.066	0.032	0.017
$\pm 3.75\%$	CI Coverage (%)	90.7	91.3	91.8	90.0	96.1	96.7	95.9	94.7
	Avg. sample size	893,860	965,410	929,527	798,234	1,205,400	1,344,800	1,337,112	1,115,986
	Avg. CI half-length	0.281	0.273	0.286	0.287	0.288	0.278	0.286	0.288
	StDev. CI half-length	0.023	0.022	0.014	0.017	0.016	0.018	0.014	0.014

Table 14 reports performance metrics for SPOSTS versus SPSTS, Skart, and ASAP3 under the three precision conditions. For the specific process under consideration, the CI coverage probabilities for SPOSTS always met or slightly exceeded the user-specified nominal levels. Under the no-precision requirement, the average sample sizes required by SPOSTS were the same as for SPSTS and substantially larger than Skart and ASAP3, but the latter two procedures failed to meet the user-specified nominal levels. Under the 7.5% relative precision requirement, the sample sizes and the CI coverage probabilities of SPOSTS were similar to those of SPSTS, and substantially higher than both Skart and ASAP3. Under the more-stringent 3.75% relative precision requirement, SPOSTS slightly outperformed SPSTS (but not Skart) in terms of the average sample size.

Table 14: Experimental results for the SPOSTS algorithm for an $M/M/1/$*LIFO* system with traffic intensity 0.8

Precision	Performance Measure	Nominal 90% CIs				Nominal 95% CIs			
		SPOSTS	SPSTS	Skart	ASAP3	SPOSTS	SPSTS	Skart	ASAP3
none	CI Coverage (%)	93.7	94.2	85.6	87.0	97.5	98.5	92.6	92.5
	Avg. sample size	143,185	143,185	21,176	53,958	143,185	143,185	21,176	53,958
	Avg. CI half-length	0.177	0.180	0.514	0.106	0.212	0.216	0.699	0.312
	StDev. CI half-length	0.045	0.039	0.296	0.106	0.054	0.047	0.247	0.008
$\pm 7.50\%$	CI Coverage (%)	93.6	93.9	91.6	87.5	97.3	98.1	95.9	92.5
	Avg. sample size	145,104	144,490	81,441	68,325	152,399	153,182	122,391	90,911
	Avg. CI half-length	0.174	0.177	0.224	0.219	0.198	0.202	0.222	0.226
	StDev. CI half-length	0.039	0.035	0.017	0.022	0.035	0.031	0.014	0.017
$\pm 3.75\%$	CI Coverage (%)	91.3	93.3	91.6		96.7	96.5	96.4	
	Avg. sample size	337,204	374,815	305,903		469,445	516,265	444,852	
	Avg. CI half-length	0.107	0.104	0.114		0.108	0.106	0.114	
	StDev. CI half-length	0.012	0.014	< 0.01		0.012	0.012	< 0.01	

4.4.4 AR(1)-to-Pareto (ARTOP) Process

The last testbed for the performance of SPOSTS was the AR(1)-to-Pareto process described in Section 3.4.4 with location parameter $\xi = 1.0$ and shape parameter $\psi = 2.1$. The marginal distribution of the ARTOP process has mean $\mu = 1.9091$, variance $\sigma_X^2 = 17.3554$, and infinite skewness and kurtosis.

We put SPOSTS on equal footing with SPSTS and Skart by initializing the "base" AR(1) process in a state with a value of 3.4. This yields an initial state with the value of 43.5689 for the ARTOP process, which is 10 standard deviations above the mean μ. On the other hand, ASAP3 was initialized in steady state.

The performance of SPOSTS in this challenging instance was similar to the previous three cases. Under the no-precision requirement, the results of SPOSTS were similar to those obtained by SPSTS. Under the 7.5% and 3.75% relative precision requirements, the estimated CI coverage probabilities were remarkably close to those of SPSTS and Skart, and significantly better than ASAP3's. Most importantly, SPOSTS required smaller sample sizes, on average, than SPSTS and Skart to achieve the same nominal precision levels.

Table 15: Experimental results for the SPOSTS algorithm for an ARTOP Process

Precision	Performance Measure	Nominal 90% CIs				Nominal 95% CIs			
		SPOSTS	SPSTS	Skart	ASAP3	SPOSTS	SPSTS	Skart	ASAP3
none	CI Coverage (%)	90.2	90.3	88.3	85.5	93.0	93.0	93.5	90.8
	Avg. sample size	306,590	306,590	37,923	114,053	306,590	306,590	37,923	114,053
	Avg. CI half-length	0.114	0.112	0.640	0.173	0.138	0.136	1.030	0.207
	StDev. CI half-length	0.087	0.028	0.788	0.099	0.100	0.033	2.430	0.120
$\pm 7.50\%$	CI Coverage (%)	90.0	89.3	88.3	84.0	93.0	92.7	95.7	90.3
	Avg. sample size	326,760	322,760	333,666	186,517	345,130	362,270	478,926	255,512
	Avg. CI half-length	0.106	0.110	0.122	0.127	0.120	0.122	0.121	0.131
	StDev. CI half-length	0.022	0.021	0.017	0.014	0.018	0.017	0.017	0.010
$\pm 3.75\%$	CI Coverage (%)	88.7	89.0	91.1	88.8	95.1	95.3	96.6	91.0
	Avg. sample size	903,200	1,035,300	1,098,130	734,312	1,294,200	1,434,100	1,588,612	1,044,259
	Avg. CI half-length	0.064	0.062	0.064	0.067	0.064	0.063	0.064	0.067
	StDev. CI half-length	0.007	0.008	< 0.01	< 0.01	0.007	0.007	< 0.01	< 0.01

4.5 Conclusions

This chapter introduced SPOSTS, the first sequential procedure for computing valid point and CI estimators for the steady-state mean of a simulation output process based solely on overlapping variance estimators. In an extensive empirical study involving three queueing

systems and an ARTOP process, we found that SPOSTS performed well compared with other sequential procedures such as Skart, ASAP3 and SPSTS in terms of CI coverage, average required sample size, and the first two moments of the CI's half-length. The most noticeable trait of SPOSTS in comparison with SPSTS was the approximately 8%–10% reduction of the sample size required to deliver CIs with coverage probabilities very close to the nominal levels under stringent precision requirements.

CHAPTER V

SUMMARY AND RECOMMENDATIONS FOR FUTURE WORK

5.1 Summary of Research Contributions

This dissertation made three major contributions. Chapter 2 focused on robust estimation for Gaussian graphical models. We proposed an iteratively weighted graphical lasso estimation algorithm by adopting the nonparametric ISE criterion and a majorization-minimization technique. Compared to traditional methods, the results of simulation studies suggested that the proposed method is more robust to anomalous observations.

In Chapters 3 and 4, we developed two automated sequential procedures (SPSTS and SPOSTS) for computing point and CI estimators for the steady-state mean of a simulation output process. SPSTS is based on nonoverlapping STS area estimators while SPOSTS is based on overlapping versions of the estimators. Empirical simulation studies of three queueing systems and an ARTOP process showed that both SPSTS and SPOSTS performed well against state-of-the-art methods based on nonoverlapping batch means.

In the next section, we describe two directions for further research work along this line.

5.2 Recommendations for Future Work

5.2.1 Sequential Estimation of σ^2

As discussed earlier, Alexopoulos et al. (2007a) studied approximations for a variety of estimators of σ^2 by appropriately scaled χ^2 random variables; see, e.g. Equations (3.2.7), (4.2.4), and (4.2.6). These approximations can be used for constructing approximate CIs for the variance parameter σ^2. For example, if m is sufficiently large and $\mathcal{A}^{\mathrm{O}}(f; b, m)$ is a first-order unbiased estimator for σ^2, then it follows immediately from Equation (4.2.4)

that an approximate $100(1-\alpha)\%$ CI for σ^2 is given by

$$\frac{\nu \mathcal{A}^{\mathrm{O}}(f;b,m)}{\chi^2_{\nu,1-\alpha/2}} \leq \sigma^2 \leq \frac{\nu \mathcal{A}^{\mathrm{O}}(f;b,m)}{\chi^2_{\nu,\alpha/2}}, \tag{5.2.1}$$

where $\chi^2_{\nu,\beta}$ denotes the β-quantile of the χ^2_ν distribution. The validity of (5.2.1) was evaluated in Section 4.3 of Alexopoulos et al. (2007a).

The challenge we plan to address in the near future is the development of automated sequential procedures for computing asymptotically valid CIs for σ^2. Preliminary experimentation indicates that CIs computed during the final stage of SPOSTS under a stringent precision requirement ($r^* \leq 3.75\%$) for the point estimate of μ exhibit coverage near the nominal value. However, it has become apparent that effective sequential procedures for estimating σ^2 will require different batching strategies than SPSTS or SPOSTS and substantially larger sample sizes.

In addition, we wish to develop simultaneous CIs for the vector (μ, σ^2).

5.2.2 Batched Cramér–von Mises (CvM) Estimators

CvM estimators are based on the weighted area under the square of the STS (Alexopoulos et al. 2007b). The *(nonoverlappping) batched CvM* estimator for σ^2 is

$$\mathcal{C}(g;b,m) \equiv \frac{1}{bm} \sum_{i=1}^{b} \sum_{\ell=1}^{m} g(\tfrac{\ell}{m})[\sigma T_{i,m}(\tfrac{\ell}{m})]^2,$$

where the weight function $g(\cdot)$ has a continuous second derivative and is normalized so that $\int_0^1 g(t)t(1-t)\,dt = 1$. Let $G \equiv \int_0^1 g(s)\,ds$. Under additional conditions for the process $X = \{X_1, X_2, \ldots\}$, one has

$$\mathrm{E}[\mathcal{C}(g;b,m)] = \sigma^2 - (G-1)\frac{\gamma_1}{m} + O(1/m^2).$$

Common choices of weights are: $g_0(t) = 6$, $g_2^\star(t) = -24 + 150t - 150t^2$, and

$$g_4^\star(t) = -\frac{1310}{21} + \frac{19270}{21}t - \frac{25230}{7}t^2 + \frac{16120}{3}t^3 - \frac{8060}{3}t^4 \quad \text{for } t \in [0,1].$$

The first estimator has first-order bias $5\gamma_1/m$, while those using $g_2^\star$ and $g_4^\star$ are first-order unbiased.

Expressions for the asymptotic bias and variance for $\mathcal{C}(g; b, m)$ are displayed in Table 1 of Alexopoulos et al. (2007b). Unlike the area estimators, the choice of weight g affects the asymptotic variance of $\mathcal{C}(g; b, m)$ and the variance of its limiting functional $\mathcal{C}(g) \equiv \int_0^1 g(t)[\sigma\mathcal{B}(t)]^2\,dt$.

A promising direction for extending SPSTS to use CvM variance estimators could be based on the following analogue of a signed "area" for $\mathcal{C}(g; b, m)$:

$$\begin{aligned}\tilde{Z}_i(g; m) &= \frac{1}{m}\sum_{l=1}^{m} g(\tfrac{l}{m})\left[\sigma T_{i,m}(\tfrac{l}{m})\right] \\ &= \frac{1}{m^{3/2}}\sum_{l=1}^{m} g(\tfrac{l}{m}) l\left(\overline{X}_{i,m} - \overline{X}_{i,l}\right) \quad \text{for } i = 1, \ldots, b.\end{aligned}$$

In particular, a batch size m sufficiently large to ensure that the $\{\tilde{Z}_i(g; m) : i = 1, \ldots, b\}$ are approximately i.i.d. $\sigma N(0, 1)$ should be large enough to ensure the approximate validity of an approximation to the distribution of $\mathcal{C}(g; b, m)$ similar to that detailed in Section 5.2.1; and such a result could be used to formulate new versions of SPSTS and SPOSTS.

5.2.3 Computational Complexity

As we mentioned in Section 3.2.2, the algorithms in the LABATCH.2 suite are the only known algorithms with linear time complexity and sublinear $[O(\log_2 n)]$ space complexity. Alexopoulos et al. (2007b) showed that for fixed n, the area and CvM estimators can be computed in $O(n)$ time and $O(m)$ space. While the latter is an upper bound per iteration, the overall time complexity for Algorithms SPSTS and SPOSTS remains an open problem that we plan to address.

APPENDIX A

CONVEX RELAXATION OF THE OBJECTIVE FUNCTION (2.3.3)

Let $\mathbf{X} \sim N_p(\mathbf{0}, \mathbf{\Sigma})$. It is well known that $\mathbf{X}\mathbf{X}^T \sim W_p(\mathbf{\Sigma}, 1)$, where W_p is the Wishart distribution. We have $\mathbf{\Sigma}^{-1/2}\mathbf{X}\mathbf{X}^T\mathbf{\Sigma}^{-1/2} \sim W_p(\mathbf{I}, 1)$. Further, $\mathrm{E}[\mathbf{\Sigma}^{-1/2}\mathbf{X}\mathbf{X}^T\mathbf{\Sigma}^{-1/2}] = \mathbf{I}_{p\times p}$, so that $\mathrm{tr}\left(\mathrm{E}[\mathbf{\Sigma}^{-1/2}\mathbf{X}\mathbf{X}^T\mathbf{\Sigma}^{-1/2}]\right) = p$, which leads to $\mathrm{E}[\mathrm{tr}\left(\mathbf{\Sigma}^{-1/2}\mathbf{X}\mathbf{X}^T\mathbf{\Sigma}^{-1/2}\right)] = p$. Thus, we have

$$\mathrm{E}[\mathbf{X}^T\mathbf{\Sigma}^{-1}\mathbf{X}] = \mathrm{E}[\mathrm{tr}\left(\mathbf{X}^T\mathbf{\Sigma}^{-1/2}\mathbf{\Sigma}^{-1/2}\mathbf{X}\right)] = \mathrm{E}[\mathrm{tr}\left(\mathbf{\Sigma}^{-1/2}\mathbf{X}\mathbf{X}^T\mathbf{\Sigma}^{-1/2}\right)] = p.$$

We can approximate $\frac{1}{2}\mathbf{x}_i^T\mathbf{\Omega}\mathbf{x}_i \cong \frac{p}{2}$. Then we have

$$\sum_{i=1}^{n} \exp\left(-\tfrac{1}{2}\mathbf{x}_i^T\mathbf{\Omega}\mathbf{x}_i\right) \cong ne^{-p/2}.$$

Clearly,

$$\sup\left\{\sum_{i=1}^{n} \exp\left(-\tfrac{1}{2}\mathbf{x}_i^T\mathbf{\Omega}\mathbf{x}_i\right) : \mathbf{x}_i \in \mathbf{R}^p \text{ for } i = 1, 2, \ldots, n\right\} = n;$$

and therefore we see that

$$\frac{1}{2^p\pi^{p/2}}\left(1 - \frac{2}{n}2^{p/2}\sum_{i=1}^{n} \exp\left(-\frac{1}{2}\mathbf{x}_i^T\mathbf{\Omega}\mathbf{x}_i\right)\right) \geq \beta,$$

where

$$\beta = \frac{1}{2^p\pi^{p/2}}\left(1 - 2^{1+(p/2)}\right) = -\frac{2}{(2\pi)^{p/2}} + \frac{1}{2^p\pi^{p/2}}.$$

This inequality can be expressed as a convex constraint $f(\mathbf{\Omega}) \leq 0$, where

$$f(\mathbf{\Omega}) = \beta - \frac{1}{2^p\pi^{p/2}}\left(1 - \frac{2}{n}2^{p/2}\sum_{i=1}^{n} \exp\left(-\frac{1}{2}\mathbf{x}_i^T\mathbf{\Omega}\mathbf{x}_i\right)\right).$$

Hence,

$$\begin{aligned} L_n(\mathbf{\Omega}) &\propto |\mathbf{\Omega}|^{1/2}\left[2^{-p/2} - \frac{2}{n}\sum_{i=1}^{n} \exp\left(-\frac{1}{2}\mathbf{x}_i^T\mathbf{\Omega}\mathbf{x}_i\right)\right] \\ &\approx |\mathbf{\Omega}|^{1/2}\left[2^{-p/2} - 2e^{-p/2}\right] \end{aligned}$$

$$\propto |\mathbf{\Omega}|^{1/2}.$$

The inverse covariance matrix $\mathbf{\Omega}$ needs to be positive definite, denoted as $\mathbf{\Omega} \succ \mathbf{0}$. This means that for large p, we can convert the problem of solving objective function (2.3.3) in the following way.

$$\begin{aligned} & \min_{\mathbf{\Omega}} |\mathbf{\Omega}|^{1/2} + \rho_n \|\mathbf{\Omega}\|_1 \\ \text{subject to} \quad & f(\mathbf{\Omega}) \leq 0 \\ & \mathbf{\Omega} \succ \mathbf{0} \end{aligned}$$

However, the objective function of this optimization problem is still nonconvex, so we need to proceed with further transformation. Let $\log |\mathbf{\Omega}| \leq t$, where t is some upper bound; then the problem can be converted to

$$\begin{aligned} & \min_{\mathbf{\Omega}} e^{t/2} + \rho_n \|\mathbf{\Omega}\|_1 \\ \text{subject to} \quad & \log |\mathbf{\Omega}| \leq t \\ & f(\mathbf{\Omega}) \leq 0 \\ & \mathbf{\Omega} \succ \mathbf{0} \end{aligned}$$

To solve this constrained optimization problem, we set an initial symmetric positive definite matrix $\mathbf{\Omega}_0$ satisfying the three constraints. We can get t_1, the value of t for the first iteration, by inputting the value of $\mathbf{\Omega}_0$ and solving the above constrained optimization problem. Then we can obtain $\mathbf{\Omega}_1$ by incorporating t_1 into the minimization problem. Doing this iteratively until the stopping criterion is met. When the Frobenius squared distance between the two consecutive estimators is smaller than 10^{-6}, we derive the final value of the estimator $\widehat{\mathbf{\Omega}}$. The entire procedure can be programmed using the CVX package in Matlab (`http://cvxr.com/cvx/`).

APPENDIX B

PROOF OF LEMMA 2.3.1

Proof. Suppose that the eigenvalues of $\mathbf{\Omega}$ are d_k, $k = 1, \ldots, p$. Using the arithmetic-geometric mean inequality, we have $\sqrt[p]{\prod_{k=1}^{p} d_i} \leq \frac{1}{p} \sum_{k=1}^{p} d_i$, and then

$$|\mathbf{\Omega}|^{1/2} \leq \left(\frac{\sum_{k=1}^{p} d_i}{p} \right)^{p/2}. \tag{B.1}$$

For each eigenvalue d_i, we apply Gershgorin's Circle Theorem (Dattorro 2005) to obtain its upper bound,

$$|d_i - c_{kk}| \leq \sum_{\substack{j=1 \\ j \neq k}}^{p} |c_{ij}| \text{ for some } k \in \{1, \ldots, p\}; \tag{B.2}$$

and it follows that

$$|d_i| \leq \max_{1 \leq k \leq p} \{ \sum_{j=1}^{p} |c_{kj}| \} \leq \sum_{k=1}^{p} \sum_{j=1}^{p} |c_{kj}| = \|\mathbf{\Omega}\|_1 \text{ for } i = 1, \ldots, p. \tag{B.3}$$

Combining (B.1) and (B.3), we have

$$|\mathbf{\Omega}|^{1/2} \leq \|\mathbf{\Omega}\|_1^{p/2}.$$

□

APPENDIX C

DERIVATION OF THE WEIGHTS IN (2.3.6)

For notational convenience, denote $g(\mathbf{\Omega}) \equiv \log\left[\frac{1}{n}\sum_{i=1}^{n}\exp(-\frac{1}{2}\mathbf{x}_i^T\mathbf{\Omega}\mathbf{x}_i)\right]$. Given an initial estimate $\mathbf{\Omega}_0$ of $\mathbf{\Omega}^*$, we apply the mean-value theorem to the function $g(\mathbf{\Omega})$ about $\mathbf{\Omega}_0$, which is given explicitly by

$$g(\mathbf{\Omega}) = g(\mathbf{\Omega}_0) + \sum_{k=1}^{p}\sum_{l=k}^{p}\left[\frac{\partial g(\mathbf{\Omega})}{\partial\Omega_{kl}}\bigg|_{\mathbf{\Omega}=\mathbf{\Omega_1}}\right](\mathbf{\Omega}-\mathbf{\Omega}_0)_{kl}\,, \tag{C.1}$$

where $\partial g(\mathbf{\Omega})/\partial\Omega_{kl}$ denotes the usual derivative of the function $g(\mathbf{\Omega})$ with respect to the (k,l) element $\mathbf{\Omega}_{kl}$ of $\mathbf{\Omega}$ for $1 \le k \le l \le p$; see Equations (1846) and (1847) of Dattorro (2005).

By the chain rule for ordinary derivatives, we have for $1 \le k \le l \le p$,

$$\begin{aligned}\frac{\partial g(\mathbf{\Omega})}{\partial\Omega_{kl}} &= \frac{\frac{\partial}{\partial\Omega_{kl}}\left[\frac{1}{n}\sum_{i=1}^{n}\exp\left(-\frac{1}{2}\mathbf{x}_i^T\mathbf{\Omega}\mathbf{x}_i\right)\right]}{\frac{1}{n}\sum_{i=1}^{n}\exp\left(-\frac{1}{2}\mathbf{x}_i^T\mathbf{\Omega}\mathbf{x}_i\right)} \\ &= \frac{-\frac{1}{2n}\sum_{i=1}^{n}\left\{\exp\left(-\frac{1}{2}\mathbf{x}_i^T\mathbf{\Omega}\mathbf{x}_i\right)\left[\frac{\partial}{\partial\Omega_{kl}}\left(\mathbf{x}_i^T\mathbf{\Omega}\mathbf{x}_i\right)\right]\right\}}{\frac{1}{n}\sum_{i=1}^{n}\exp\left(-\frac{1}{2}\mathbf{x}_i^T\mathbf{\Omega}\mathbf{x}_i\right)}\end{aligned} \tag{C.2}$$

Now because $\mathbf{\Omega}$ must be symmetric, by Exercise 8(b)(2) in Chapter 15 of Harville (2008), we must have

$$\frac{\partial}{\partial\Omega_{kl}}\left(\mathbf{x}_i^T\mathbf{\Omega}\mathbf{x}_i\right) = \begin{cases} x_{ik}^2 & \text{if } 1 \le k = l \le p, \\ 2x_{ik}x_{il} & \text{if } 1 \le k < l \le p. \end{cases} \tag{C.3}$$

Evaluating (C.2) and (C.3) at $\mathbf{\Omega} = \mathbf{\Omega}_1$, and after some algebra, we have for $1 \le k \le l \le p$,

$$\frac{\partial g(\mathbf{\Omega})}{\partial\Omega_{kl}} = \begin{cases} -\frac{1}{2n}\sum_{i=1}^{n} w_i(\mathbf{\Omega}_1)x_{ik}^2 & \text{if } 1 \le k = l \le p, \\ -\frac{1}{n}\sum_{i=1}^{n} w_i(\mathbf{\Omega}_1)x_{ik}x_{il} & \text{if } 1 \le k \le l \le p, \end{cases} \tag{C.4}$$

where $w_i \equiv w_i(\mathbf{\Omega}_1)$ is given by

$$w_i \equiv w_i(\mathbf{\Omega}_1) = \frac{\exp\left(-\frac{1}{2}\mathbf{x}_i^T \mathbf{\Omega}_1 \mathbf{x}_i\right)}{\frac{1}{n}\sum_{j=1}^{n} \exp\left(-\frac{1}{2}\mathbf{x}_j^T \mathbf{\Omega}_1 \mathbf{x}_j\right)}, \text{for } i = 1, 2, \ldots, n. \tag{C.5}$$

Inserting (C.4) and (C.5) into (C.1), we have

$$\begin{aligned} g(\mathbf{\Omega}) &= g(\mathbf{\Omega}_0) + \sum_{k=1}^{p} \left[\frac{\partial g(\mathbf{\Omega})}{\partial \mathbf{\Omega}_{kk}}\bigg|_{\mathbf{\Omega}=\mathbf{\Omega_1}}\right] (\mathbf{\Omega} - \mathbf{\Omega}_0)_{kk} \\ &\quad + \sum_{k=1}^{p-1}\sum_{l=k+1}^{p} \left[\frac{\partial g(\mathbf{\Omega})}{\partial \mathbf{\Omega}_{kl}}\bigg|_{\mathbf{\Omega}=\mathbf{\Omega_1}}\right] (\mathbf{\Omega} - \mathbf{\Omega}_0)_{kl} \\ &= g(\mathbf{\Omega}_0) - \frac{1}{2n}\sum_{k=1}^{p}\sum_{i=1}^{n} w_i(\mathbf{\Omega}_1) x_{ik}^2 (\mathbf{\Omega} - \mathbf{\Omega}_0)_{kk} \\ &\quad - \frac{1}{n}\sum_{k=1}^{p-1}\sum_{l=k+1}^{p}\sum_{i=1}^{n} w_i(\mathbf{\Omega}_1) x_{ik} x_{il} (\mathbf{\Omega} - \mathbf{\Omega}_0)_{kl} \\ &= g(\mathbf{\Omega}_0) - \frac{1}{2n}\sum_{i=1}^{n} w_i(\mathbf{\Omega}_1) \left\{\sum_{k=1}^{p}\sum_{l=1}^{p} x_{ik} x_{il} (\mathbf{\Omega} - \mathbf{\Omega}_0)_{kl}\right\}. \end{aligned} \tag{C.6}$$

In order to simplify (C.6), we first prove four claims:

Claim 1: For two $p \times p$ matrices $\boldsymbol{U}$, $\boldsymbol{V}$, we have $\mathrm{tr}\,(\boldsymbol{UV}) = \sum_{k=1}^{p}\sum_{l=1}^{p} u_{lk} v_{kl}$.

Claim 2: the (l, k) entry of $\mathbf{x}_i \mathbf{x}_i^T$ is $x_{il} x_{ik}$.

Claim 3: $\mathrm{tr}\left(\mathbf{x}_i \mathbf{x}_i^T (\mathbf{\Omega} - \mathbf{\Omega}_0)\right) = \sum_{k=1}^{p}\sum_{l=1}^{p} x_{il} x_{ik} (\mathbf{\Omega} - \mathbf{\Omega}_0)_{kl}$.

Claim 4: $\mathrm{tr}\,(\boldsymbol{A}(\mathbf{\Omega} - \mathbf{\Omega}_0)) = -\frac{1}{2n}\sum_{i=1}^{n} w_i \{\sum_{k=1}^{p}\sum_{l=1}^{p} x_{ik} x_{il} (\mathbf{\Omega} - \mathbf{\Omega}_0)\}$.

Proof. The proofs for Claims 1 and 2 are trivial. Taking $\boldsymbol{U} = \mathbf{x}_i \mathbf{x}_i^T$, $\boldsymbol{V} = \mathbf{\Omega} - \mathbf{\Omega}_0$ in Claim 1, and applying Claim 2, we have Claim 3.

The proof for Claim 4 is as follows: define $\boldsymbol{A} \equiv -\frac{1}{2n}\sum_{i=1}^{n} w_i \mathbf{x}_i \mathbf{x}_i^T$. By the linearity of the trace, we have

$$\mathrm{tr}\,(\boldsymbol{A}(\mathbf{\Omega} - \mathbf{\Omega}_0)) = -\frac{1}{2n}\sum_{i=1}^{n} w_i \mathrm{tr}\left(\mathbf{x}_i \mathbf{x}_i^T (\mathbf{\Omega} - \mathbf{\Omega}_0)\right).$$

□

By Claims 1–4, we have

$$-\frac{1}{2n}\sum_{i=1}^{n} w_i \mathrm{tr}\left(\mathbf{x}_i \mathbf{x}_i^T (\mathbf{\Omega} - \mathbf{\Omega}_0)\right) = \mathrm{tr}\,(\boldsymbol{A}(\mathbf{\Omega} - \mathbf{\Omega}_0)). \tag{C.7}$$

From the above, we are able to derive the Taylor expansion as

$$g(\mathbf{\Omega}) = g(\mathbf{\Omega}_0) + \mathrm{tr}[\mathbf{A}(\mathbf{\Omega} - \mathbf{\Omega_0})],$$

with $\mathbf{A} = -\frac{1}{2n}\sum_{i=1}^{n} w_i \mathbf{x}_i \mathbf{x}_i^T$.

APPENDIX D

THE 26 KEY GENES USED IN SECTION 2.5.1

Gene Symbol	Gene Name
AMFR	Autocrine motility factor receptor
BBS4	Bardet-Biedl syndrome 4
BECN1	Beclin 1 (coiled-coil, myosin-like BCL2 interacting protein)
BTG3	BTG family, member 3
CA12	Carbonic anhydrase XII
CTNND2	Catenin, delta 2
E2F3	E2F transcription factor 3
ERBB4	Verba erythroblastic leukemia viral oncogene homolog 4 (avian)
FGFR1OP	FGFR1 oncogene partner
FLJ10916	Hypothetical protein FLJ10916
FLJ12650	Hypothetical protein FLJ12650
GAMT	Guanidinoacetate N-methyl transferase
GFRA1	GDNF family receptor 1
IGFBP4	Insulin-like growth factor binding protein 4
JMJD2B	Jumonji domain containing 2B
KIAA1467	KIAA1467 protein
MAPT	Microtubule-associated protein
MBTP-S1	Hypothetical protein
MELK	Maternal embryonic leucine zipper kinase
MTRN	Meteorin, glial cell differentiation regulator
PDGFRA	Human clone 23, 948 mRNA sequence
RAMP1	Receptor (calcitonin) activity modifying protein 1
RRM2	Ribonucleotide reductase M2 polypeptide
SCUBE2	Signal peptide, CUB domain EGF-like 2
THRAP2	Thyroid hormone receptor associated protein 2
ZNF552	Zinc finger protein 552

APPENDIX E

PROOF OF THEOREM 3.2.1

Proof. The proof is parallel to Theorem 3 and the corresponding examples in Aktaran-Kalaycı et al. (2007). By following Goldsman (1999), the expected value of the estimator can be written as

$$\mathrm{E}[\mathcal{A}^{\mathrm{O}}(f_{\cos,j};b,m)] \;=\; \frac{1}{m^3}\left(R_0\sum_{t=1}^{m}h^2(t)+2\sum_{i=1}^{m-1}R_i\sum_{t=1}^{m-i}h(t)h(t+i)\right), \tag{E.1}$$

where $h(t) \;\equiv\; \sum_{k=1}^{m}\frac{k}{m}f(\frac{k}{m})-\sum_{k=t}^{m}f(\frac{k}{m}), \quad t=1,2,\ldots,m.$

First, we have

$$\sum_{k=1}^{m}k\cos(k\theta)=\frac{m\sin(\frac{(2m+1)\theta}{2})}{2\sin(\frac{\theta}{2})}+\frac{\cos(m\theta)-1}{4\sin^2\left(\frac{\theta}{2}\right)} \tag{E.2}$$

and

$$\sum_{k=1}^{m}\cos(k\theta)=\frac{\sin(\frac{(2m+1)\theta}{2})}{2\sin\left(\frac{\theta}{2}\right)}-\frac{1}{2}. \tag{E.3}$$

By Equation (E.2), we can compute the first term of $h(t)$,

$$\begin{aligned}\sum_{k=1}^{m}\tfrac{k}{m}f(\tfrac{k}{m}) &= \frac{\sqrt{8}\pi j}{m}\sum_{k=1}^{m}k\cos(\tfrac{k2\pi j}{m})\\ &= \frac{\sqrt{8}\pi j}{m}\left(\frac{m\sin(\frac{\pi j(2m+1)}{m})}{2\sin(\frac{\pi j}{m})}+\frac{\cos(2\pi j)-1}{4\sin^2\left(\frac{\pi j}{m}\right)}\right)\\ &= \sqrt{2}\pi j.\end{aligned} \tag{E.4}$$

By Equation (E.3), we can simplify the second term of $h(t)$,

$$\begin{aligned}\sum_{k=t}^{m}f(\tfrac{k}{m}) &= \sum_{k=1}^{m}f(\tfrac{k}{m})-\sum_{k=1}^{t-1}f(\tfrac{k}{m})\\ &= \sqrt{8}\pi j\left(\frac{\sin\left(\frac{\pi j(2m+1)}{m}\right)}{2\sin(\frac{\pi j}{m})}-\frac{1}{2}\right)-\sqrt{8}\pi j\left(\frac{\sin(\frac{\pi j(2t-1)}{m})}{2\sin(\frac{\pi j}{m})}-\frac{1}{2}\right)\end{aligned}$$

$$= -\sqrt{2}\pi j \sin(\tfrac{\pi j(2t-1)}{m}) \csc(\tfrac{\pi j}{m}) + \sqrt{2}\pi j. \tag{E.5}$$

Combining Equations (E.4) and (E.5), we have a closed form for $h(t)$,

$$\begin{aligned} h(t) &= \sqrt{2}\pi j + \sqrt{2}\pi j \sin(\tfrac{\pi j(2t-1)}{m}) \csc(\tfrac{\pi j}{m}) - \sqrt{2}\pi j \\ &= \sqrt{2}\pi j \sin(\tfrac{\pi j(2t-1)}{m}) \csc(\tfrac{\pi j}{m}). \end{aligned} \tag{E.6}$$

Continuing, we have

$$\begin{aligned} \sum_{t=1}^{m} h^2(t) &= 2\pi^2 j^2 \csc^2(\tfrac{\pi j}{m}) \sum_{t=1}^{m} \sin^2(\tfrac{\pi j(2t-1)}{m}) \\ &= 2\pi^2 j^2 \csc^2(\tfrac{\pi j}{m}) \sum_{t=1}^{m} \left(\frac{1}{2} - \tfrac{1}{2}\cos(\tfrac{2\pi j(2t-1)}{m}) \right) \\ &= m\pi^2 j^2 \csc^2(\tfrac{\pi j}{m}) - \pi^2 j^2 \csc^2(\tfrac{\pi j}{m}) \sin(2\pi t j) \csc(\tfrac{2\pi j}{m}) \cos(2\pi j) \\ &= m\pi^2 j^2 \csc^2(\tfrac{\pi j}{m}). \end{aligned} \tag{E.7}$$

In addition,

$$\begin{aligned} \sum_{t=1}^{m-i} h(t)h(t+i) &= 2\pi^2 j^2 \csc^2(\tfrac{\pi j}{m}) \sum_{t=1}^{m-i} \sin(\tfrac{\pi j(2t-1)}{m}) \sin(\tfrac{\pi j(2t+2i-1)}{m}) \\ &= \pi^2 j^2 \csc^2(\tfrac{\pi j}{m}) \sum_{t=1}^{m-i} \left(\cos(\tfrac{2\pi j i}{m}) - \cos(\tfrac{\pi j(4t+2i-2)}{m}) \right) \\ &= \pi^2 j^2 \csc^2(\tfrac{\pi j}{m}) \left\{ (m-i)\cos(\tfrac{2\pi j i}{m}) - \sum_{t=1}^{m-i} \cos(\tfrac{\pi j(4t+2i-2)}{m}) \right\} \\ &= \pi^2 j^2 \csc^2(\tfrac{\pi j}{m}) \left\{ (m-i)\cos(\tfrac{2\pi j i}{m}) + \sin(\tfrac{2\pi j i}{m}) \csc(\tfrac{2\pi j}{m}) \right\}. \end{aligned}$$

Putting (E.7) and (E.8) into (E.1), we obtain

$$\begin{aligned} &\mathrm{E}[\mathcal{A}^{\mathrm{O}}(f_{\cos,j}; b, m)] \\ &= \frac{\pi^2 j^2 \csc^2(\frac{\pi j}{m})}{m^2} \left\{ R_0 + \frac{2}{m} \sum_{i=1}^{m-1} R_i \left[(m-i)\cos(\tfrac{2\pi j i}{m}) + \sin(\tfrac{2\pi j i}{m}) \csc(\tfrac{2\pi j}{m}) \right] \right\}. \end{aligned}$$

By Talyor expansion, we have

$$\sin(x) = x - \frac{1}{6}x^3 + \frac{1}{120}x^5 + \sum_{n=3}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!}, \tag{E.8}$$

$$\cos(x) = 1 - \frac{1}{2}x^2 + \frac{1}{24}x^4 + \sum_{n=3}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!}, \tag{E.9}$$

$$\csc(x) = \frac{1}{x} + \frac{1}{6}x + \frac{7}{360}x^3 + \sum_{n=3}^{\infty} \frac{(-1)^{n+1}(2^{2n}-2)B_{2n}x^{2n-1}}{(2n)!}, \tag{E.10}$$

where the B_{2n} are Bernoulli numbers, and $B_{2n} \approx (-1)^{n-1}4\sqrt{\pi n}\left(\frac{n}{\pi e}\right)^{2n}$ (Digital Library of Mathematical Functions (NIST), Section 24.11, `http://dlmf.nist.gov/24`).

By Stirling's approximation, we have $n! \approx \left(\frac{n}{e}\right)^n \sqrt{2\pi n}$; thus,

$$\frac{(-1)^{n+1}B_{2n}}{(2n)!} \approx \frac{(-1)^{2n}4\sqrt{\pi n}\left(\frac{n}{\pi e}\right)^{2n}}{\left(\frac{2n}{e}\right)^{2n}\sqrt{4\pi n}} = \frac{2}{(2\pi)^{2n}}.$$

Inserting Equations (E.8), (E.9), and (E.10) into Equation (E.8), we have

$$\begin{aligned}
\frac{\pi^2 j^2 \csc^2(\frac{\pi j}{m})}{m^2} &= \frac{\pi^2 j^2}{m^2}\left(\frac{m}{\pi j} + \frac{\pi j}{6m} + \frac{7\pi^3 j^3}{360m^3} + \sum_{n=3}^{\infty}\frac{(-1)^{n+1}(2^{2n}-2)B_{2n}x^{2n-1}}{(2n)!}\right)^2 \\
&\approx \left(1 + \frac{\pi^2 j^2}{6m^2} + \frac{7\pi^4 j^4}{360m^4} + \sum_{n=3}^{\infty}\frac{2(2^n-2)\left(\pi j m\right)^{2n}}{(2\pi)^{2n}}\right)^2 \\
&= \left(1 + \frac{\pi^2 j^2}{6m^2} + \frac{7\pi^4 j^4}{360m^4} + 2\sum_{n=3}^{\infty}\left[\left(\frac{j}{m}\right)^{2n} - 2\left(\frac{j}{2m}\right)^{2n}\right]\right)^2 \\
&= \left(1 + \frac{\pi^2 j^2}{6m^2} + \frac{7\pi^4 j^4}{360m^4} + \frac{2j^6}{m^4(m^2-j^2)} - \frac{j^6}{4m^4(4m^2-j^2)}\right)^2 \\
&= 1 + \frac{\pi^2 j^2}{3m^2} + \frac{\pi^4 j^4}{15m^4} + \frac{(31m^2-7j^2)j^6}{2m^4(m^2-j^2)(4m^2-j^2)} + O(\tfrac{1}{m^6}) \\
&= 1 + \frac{\pi^2 j^2}{3m^2} + \frac{\pi^4 j^4}{15m^4} + O(\tfrac{1}{m^6}),
\end{aligned} \tag{E.11}$$

where $\frac{(31m^2-7j^2)j^6}{2m^4(m^2-j^2)(4m^2-j^2)} = O(\frac{1}{m^6})$, since $j \le j^* \ll m$ (j^* is fixed).

We also have

$$\begin{aligned}
(m-i)\cos(\tfrac{2\pi ji}{m}) &= (m-i)\left[1 - \frac{1}{2}\left(\frac{2\pi ji}{m}\right)^2 + \frac{1}{24}\left(\frac{2\pi ji}{m}\right)^4 + \sum_{n=3}^{\infty}\frac{(-1)^n x^{2n}}{(2n)!}\right] \\
&= (m-i) - \frac{2\pi^2 j^2 i^2}{m} + \frac{2\pi^2 j^2 i^3}{m^2} + \frac{2\pi^4 j^4 i^4}{3m^3} - \frac{2\pi^4 j^4 i^5}{3m^4} \\
&\quad + (m-i)\sum_{n=3}^{\infty}\frac{(-1)^n\left(\frac{2\pi ji}{m}\right)^{2n}}{(2n)!}.
\end{aligned} \tag{E.12}$$

and

$\sin(\frac{2\pi ji}{m})\csc(\frac{2\pi j}{m})$

$$
\begin{aligned}
&= \left[\frac{2\pi ji}{m} - \frac{4\pi^3 j^3 i^3}{3m^3} + \sum_{n=3}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!}\right]\left[\frac{m}{2\pi j} + \frac{\pi j}{3m} + \frac{7\pi^3 j^3}{45m^3} + \frac{6j^5(5m^2-4j^2)}{\pi m^3(m^2-4j^2)(m^2-j^2)}\right] \\
&= i - \frac{2\pi^2 j^2 i^3}{3m^2} + \frac{2\pi^2 j^2 i}{3m^2} + \sum_{n=3}^{\infty} \frac{(-1)^n \left(\frac{2\pi ji}{m}\right)^{2n+1}}{(2n+1)!}\left(\frac{m}{2\pi j} + \frac{\pi j}{3m} + \frac{7\pi^3 j^3}{45m^3} + O\left(\tfrac{1}{m^5}\right)\right).
\end{aligned}
$$

By (E.12) and (E.13), we have

$$
\begin{aligned}
&\frac{2}{m}(m-i)\cos(\tfrac{2\pi ji}{m}) + \frac{2}{m}\sin(\tfrac{2\pi ji}{m})\csc(\tfrac{2\pi j}{m}) \\
&= 2 - \frac{4\pi^2 j^2 i}{m^2} + \frac{8\pi^2 j^2 i^3 + 4\pi^2 j^2 i}{3m^3} + \frac{4\pi^4 j^4 i^4}{3m^4} + 2\left(1 - \frac{i}{m}\right)\sum_{n=3}^{\infty} \frac{(-1)^n \left(\frac{2\pi ji}{m}\right)^{2n}}{(2n)!} \\
&\quad + \sum_{n=3}^{\infty} \frac{(-1)^n \left(\frac{2\pi ji}{m}\right)^{2n+1}}{(2n+1)!}\left(\frac{1}{\pi j} + \frac{2\pi j}{3m^2} + \frac{14\pi^3 j^3}{45m^4} + O\left(\tfrac{1}{m^6}\right)\right) + O(\tfrac{1}{m^5}).
\end{aligned}
$$

For the remaining terms, we have the general formula,

$$
\begin{aligned}
\sum_{i=1}^{m-1}\sum_{n=k}^{\infty} i^q \frac{\left(\frac{ci}{m}\right)^n}{n!}|R_i| &\approx \sum_{i=1}^{m-1} i^q \sum_{n=k}^{\infty} \frac{\left(\frac{ci}{m}\right)^n}{\left(\frac{n}{e}\right)^n\sqrt{2\pi n}}|R_i| \\
&\le \frac{1}{\sqrt{2\pi k}}\sum_{i=1}^{m-1} i^q \sum_{n=k}^{\infty}\left(\tfrac{cei}{nm}\right)^n R_i \\
&= \frac{1}{\sqrt{2\pi k}}\sum_{i=1}^{m-1} i^q \left\{\sum_{n=k}^{k'-1}\left(\tfrac{cei}{nm}\right)^n + \sum_{n=k'}^{\infty}\left(\tfrac{cei}{nm}\right)^n\right\} R_i \\
&\le \frac{1}{\sqrt{2\pi k}}\sum_{i=1}^{m-1} i^q \left\{\left(\tfrac{ce}{k}\right)^{k'-1}\sum_{n=k}^{k'-1}\left(\tfrac{i}{m}\right)^n + \sum_{n=k'}^{\infty}\left(\tfrac{i}{m}\right)^n\right\} R_i \\
&\le \frac{1}{\sqrt{2\pi k}}\left(\tfrac{ce}{k}\right)^{k'-1}\sum_{i=1}^{m-1} i^q \left\{\sum_{n=k}^{k'-1}\left(\tfrac{i}{m}\right)^n + \sum_{n=k'}^{\infty}\left(\tfrac{i}{m}\right)^n\right\} R_i \\
&= B\sum_{i=1}^{m-1} i^q \sum_{n=k}^{\infty}\left(\tfrac{i}{m}\right)^n R_i \\
&= B\sum_{i=1}^{m-1} i^q \left(\tfrac{i}{m}\right)^k \frac{m}{m-i} R_i \\
&\le \frac{B}{m^k}\sum_{i=1}^{m-1} i^{q+k} m R_i \\
&= \frac{B}{m^{k-1}}\sum_{i=1}^{m-1} i^{q+k} R_i \\
&\le \frac{2B}{m^{k-1}}\sum_{i=1}^{m-1} i^{q+k} R_i
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{B\gamma_{q+k}}{m^{k-1}} \\
&= O(\tfrac{1}{m^{k-1}}),
\end{aligned} \tag{E.13}
$$

where $B = \frac{1}{\sqrt{2\pi k}} \left(\frac{ce}{k}\right)^{k'-1}$ and k' is the first integer that satisfies $k' > ce$;. The last equation holds because $|R_i| = O(\delta^i)$, for $\delta \in (0,1)$.

Applying Equation (E.13) with $c = 2\pi j$, $q = 0$, and $k = 7$, we have

$$
2\sum_{i=1}^{m-1}\left(1-\frac{i}{m}\right)\sum_{n=3}^{\infty}\frac{(-1)^n\left(\frac{2\pi ji}{m}\right)^{2n}}{(2n)!} \leq 2\sum_{i=1}^{m-1}\sum_{n=3}^{\infty}\frac{\left(\frac{2\pi ji}{m}\right)^{2n}}{(2n)!} = O(\tfrac{1}{m^5}) \tag{E.14}
$$

and

$$
\begin{aligned}
&\sum_{i=1}^{m-1}\sum_{n=3}^{\infty}\frac{(-1)^n\left(\frac{2\pi ji}{m}\right)^{2n+1}}{(2n+1)!}\left(\frac{1}{\pi j}+\frac{2\pi j}{3m^2}+\frac{14\pi^3 j^3}{45m^4}+O\left(\tfrac{1}{m^6}\right)\right) \\
&= \sum_{i=1}^{m-1}\sum_{n=3}^{\infty}\frac{(-1)^n\left(\frac{2\pi ji}{m}\right)^{2n+1}}{(2n+1)!}\left(\frac{1}{\pi j}+O\left(\tfrac{1}{m^2}\right)\right) \\
&\leq \sum_{i=1}^{m-1}\sum_{n=3}^{\infty}\frac{\left(\frac{2\pi ji}{m}\right)^{2n+1}}{(2n+1)!} \\
&= O(\tfrac{1}{m^6}),
\end{aligned} \tag{E.15}
$$

Substituting Equations (E.11), (E.13), (E.14) and (E.15) into (E.8), we obtain

$$
\begin{aligned}
&\mathrm{E}[\mathcal{A}^{\mathrm{O}}(f_{\cos,j}; b, m)] \\
&= \left(1+\frac{\pi^2 j^2}{3m^2}+\frac{\pi^4 j^4}{15m^4}\right)\left\{R_0+2\sum_{i=1}^{m-1}R_i\left(1-\frac{2\pi^2 j^2 i^2}{m^2}+\frac{4\pi^2 j^2 i^3+2\pi^2 j^2 i}{3m^3}+\frac{2\pi^4 j^4 i^4}{3m^4}\right)\right\} \\
&\quad +O(\tfrac{1}{m^5}) \\
&= \left(1+\frac{\pi^2 j^2}{3m^2}+\frac{\pi^4 j^4}{15m^4}\right)\left\{R_0+2\sum_{i=1}^{\infty}R_i\left(1-\frac{2\pi^2 j^2 i^2}{m^2}+\frac{4\pi^2 j^2 i^3+2\pi^2 j^2 i}{3m^3}+\frac{2\pi^4 j^4 i^4}{3m^4}\right)\right\} \\
&\quad +O(\tfrac{1}{m^5}) \\
&= \left(1+\frac{\pi^2 j^2}{3m^2}+\frac{\pi^4 j^4}{15m^4}\right)\left\{\sigma^2-\frac{2\pi^2 j^2}{m^2}\gamma_2+\frac{4\pi^2 j^2\gamma_3+2\pi^2 j^2\gamma_1}{3m^3}+\frac{2\pi^4 j^4\gamma_4}{3m^4}\right\}+O(\tfrac{1}{m^5}) \\
&= \sigma^2-\frac{\pi^2 j^2(6\gamma_2-\sigma^2)}{3m^2}+\frac{2\pi^2 j^2\gamma_3}{3m^3}+\frac{\pi^4 j^4[\sigma^2+10(\gamma_4-\gamma_2)]}{15m^4}+O(\tfrac{1}{m^5}).
\end{aligned} \tag{E.16}
$$

□

Bibliography

[1] Aktaran-Kalaycı, T., C. Alexopoulos, N. T. Argon, D. Goldsman, and J. R. Wilson. 2007. "Exact Expected Values of Variance Estimators in Steady-State Simulation". *Naval Research Logistics* 54(4):397–410.

[2] Alexopoulos, C., N. T. Argon, D. Goldsman, N. M. Steiger, G. Tokol, and J. R. Wilson. 2007a. "Efficient Computation of Overlapping Variance Estimators for Simulation". *INFORMS Journal on Computing* 19(3):314–327.

[3] Alexopoulos, C., N. T. Argon, D. Goldsman, G. Tokol, and J. R. Wilson. 2007b. "Overlapping Variance Estimators for Simulation". *Operations Research* 55(6):1090–1103.

[4] Alexopoulos, C., and D. Goldsman. 2004. "To Batch or Not to Batch?". *ACM Transactions on Modeling and Computer Simulation* 14(1):76–114.

[5] Ambroise, C., J. Chiquet, and C. Matias. 2009. "Inferring Sparse Gaussian Graphical Models with Latent Structure". *Electronic Journal of Statistics* 3:205–238.

[6] Antonini, C., C. Alexopoulos, D. Goldsman, and J. R. Wilson. 2009. "Area Variance Estimators for Simulation Using Folded Standardized Time Series". *IIE Transactions* 41:134–144.

[7] Banerjee, O., L. El Ghaoui, and N. Georges. 2006. "Convex Optimization Techniques for Fitting Sparse Gaussian Graphical Models". *In Proceedings of the* $23rd$ *International Conference on Machine Learning*, 89–96.

[8] Basu, A., I. R. Harris, N. L. Hjort and M. C. Jones. 1998. "Robust and Efficient Estimation by Minimising a Density Power Divergence". *Biometrika* 85:549–559.

[9] Bickel, P. J. and E. Levina. 2008. "Regularized Estimation Of Large Covariance Matrices". *Annals of Statistics* 36(1):199–227.

[10] Box, G. E. P. 1954. "Some Theorems on Quadratic Forms Applied in the Study of Analysis of Variance Problems, I. Effect of Inequality of Variance in the One-Way Classification". *Annals of Mathematical Statistics* 25:290–302.

[11] Boyd, S., and L. Vandenberghe. 2004. *Convex Optimization*. New York: Cambridge University Press.

[12] Cai, T., W. Liu, and X. Luo. 2011. "A Constrained l_1 Minimization Approach to Sparse Precision Matrix Estimation". *Journal of American Statistical Association* 106:594–607.

[13] Chi, E. C., and W. Scott. 2012. "Robust Parametric Classification and Variable Selection by a Minimum Distance Criterion". *Journal of Computational and Graphical Statistics* 23(1):111–128.

[14] Damerdji, H. 1991. "Strong Consistency and Other Properties of the Spectral Variance Estimator". *Management Science* 37:1424–1440.

[15] Damerdji, H. 1994. "Strong Consistency of the Variance Estimator in Steady-State Simulation Output Analysis". *Mathematics of Operations Research* 19:494–512.

[16] Damerdji, H., and D. Goldsman. 1995. "Consistency of Several Variants of the Standardized Time Series Area Variance Estimator". *Naval Research Logistics* 42:1161–1176.

[17] Dattorro, J. 2005. *Convex Optimization & Euclidean Distance Geometry*. Palo Alto: Meboo Publishing.

[18] Davis, J., and M. Goadrich. 2006. "The Relationship between Precision-Recall and ROC Curves". *Proceedings of the* $23rd$ *International Conference on Machine Learning*, 233–240.

[19] Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. "Maximum Likelihood from Incomplete Data via the EM Algorithm". *Journal of the Royal Statistical Society, Series B* 39(1):1–38.

[20] Dey, D. K. 1988. "Simultaneous Estimation of Eigenvalues". *Annals of the Institute of Statistical Mathematics* 40(1):137–147.

[21] Durrett, R. 2005. *Probability: Theory and Examples*. 3rd ed. Belmont, CA: Brooks/Cole–Thomson Learning.

[22] Fan, J. and I. Gijbels. 1996. *Local Polynomial Modelling and Its Applications*. London: Chapman & Hall.

[23] Finegold, M., and M. Drton. 2011. "Robust Graphical Modeling of Gene Networks using Classical and Alternative t-distributions". *Annals of Applied Statistics* 5(2A):1057–1080.

[24] Fishman, G. S. 2001. *Discrete-Event Simulation: Modeling, Programming, and Analysis*. New York: Springer-Verlag.

[25] Fishman, G. S., and L. S. Yarberry. 1997. "An Implementation of the Batch Means Method". *INFORMS Journal on Computing* 9:296–310.

[26] Foley, R. D., and D. Goldsman. 1999. "Confidence Intervals Using Orthonormally Weighted Standardized Time Series". *ACM Transactions on Modeling and Simulation* 9:297–325.

[27] Friedman, J., T. Hastie, and R. Tibshirani. 2008. "Sparse Inverse Covariance Estimation with the Graphical Lasso". *Biostatistics* 9(3):432–441.

[28] Fuller, W. A. 1996. *Introduction to Statistical Time Series*. 2nd ed. New York: John Wiley & Sons.

[29] Glynn, P. W., and D. L. Iglehart. 1990. "Simulation Output Analysis Using Standardized Time Series". *Mathematics of Operations Research* 15:1–16.

[30] Goldsman, D., and M. S. Meketon. 1986. "A Comparison of Several Variance Estimators". *Technical Report* 12. School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA.

[31] Goldsman, D., M. S. Meketon, and L. W. Schruben. 1990. "Properties of Standardized Time Series Weighted Area Variance Estimators". *Management Science* 36:602–612.

[32] Haff, L. R. 1977. "Minimax Estimators for a Multinormal Precision Matrix". *Journal of Multivariate Analysis* 7(3):374–385.

[33] Harville, D. A. 2008. "Matrix Algebra from a Statistician's Perspective". New York: Springer.

[34] Heidelberger, P., and P. D. Welch. 1981. "A Spectral Method for Confidence Interval Generation and Run Length Control in Simulations". *Communications of the ACM* 24(4):233–245.

[35] Hess, K. R., K. Anderson, W. F. Symmans, V. Valero, N. Ibrahim, J. A. Mejia, D. Booser, R. L. Theriault, U. Buzdar, P. J. Dempsey, R. Rouzier, N. Sneige, J. S. Ross, T. Vidaurre, H. L. Gómez, G. N. Hortobagyi, and L. Pustzai. 2006. "Pharmacogenomic Predictor of Sensitivity to Preoperative Chemotherapy with Paclitaxel and Uorouracil, Doxorubicin, and Cyclophosphamide in Breast Cancer". *Journal of Clinical Oncology* 24(26):4236–4244.

[36] Horn, R. A., and C. R. Johnson.1996. *Matrix Analysis*. Cambridge, UK: Cambridge University Press.

[37] Hsieh, C. J., M. A. Sustik, S. D. Inderjit, and R. Pradeep. 2011. "Sparse Inverse Covariance Matrix Estimation Using Quadratic Approximation". *Advances in Neural Information Processing Systems* 24:2330–2338.

[38] Huang, J. Z., N. Liu, M. Pourahmadi, and L. Liu. 2006. "Covariance Matrix Selection and Estimation via Penalised Normal Likelihood". *Biometrika* 93(1):85–98.

[39] Hunter, D. R., and K. Lange. 2004. "A Tutorial on MM Algorithms". *The American Statistician* 58(1):30–37.

[40] Knight, K., and W. Fu. 2000. "Asymptotics for Lasso-Type Estimators". *The Annals of Statistics* 28(5):1356–1378.

[41] Lada, E. K., N. M. Steiger, and J. R. Wilson. 2006. "Performance Evaluation of Recent Procedures for Steady-State Analysis". *IIE Transactions* 38(9):711–727.

[42] Lada, E. K., and J. R. Wilson. 2006. "A Wavelet-Based Spectral Procedure for Steady-State Simulation Analysis". *European Journal of Operational Research* 174(3):1769–1801.

[43] Lauritzen, S. L. 1996. "Graphical Models". United Kingdom: Clarendon Press.

[44] Law, A. M. 2007. *Simulation Modeling and Analysis*. 4th ed. New York: McGraw-Hill.

[45] Law, A. M., and J. S. Carson. 1979. "A Sequential Procedure for Determining the Length of a Steady-State Simulation". *Operations Research* 27:1011–1025.

[46] Ledoit, O., and M. Wolf. 2004. "A Well-Conditioned Estimator For Large Dimensional Covariance Matrices". *Journal of Multivariate Analysis* 88(2):365–411.

[47] Levina, E., A. J. Rothman, and J. Zhu. 2008. "Sparse Estimation of Large Covariance Matrices via a Nested Lasso Penalty". *Annals of Applied Statistics* 2(1):245–263.

[48] Malkovich, J. F., and A. A. Afifi. 1973. "On Tests for Multivariate Normality". *Journal of the American Statistical Association* 68, 341(March):176–179.

[49] Markowitz, H. 1952. "Portfolio Selection". *The Journal of Finance* 7(1):77–91.

[50] Meinshausen, N. and P. Bühlmann. 2006. "High Dimensional Graphs and Variable Selection with the Lasso". *Annals of Statistics* 34(3):1436–1462.

[51] Meketon, M. S., and B. W. Schmeiser. 1984. "Overlapping Batch Means: Something for Nothing?" S. Sheppard, U. Pooch, D. Pegden, eds. *Proceedings of the 1984 Winter Simulation Conference*, Institute of Electrical and Electronics Engineers. Piscataway, NJ, 227–230.

[52] Miyamura, M., and Y. Kano. 2006. "Robust Gaussian Graphical Modeling". *Journal of Multivariate Analysis* 97(7):1525–1550.

[53] Natowicz, R., R. Incitti, E. G. Horta, B. Charles, P. Guinot, K. Yan, C. Coutant, F. André, R. Pusztai, and L. Rouzier. 2008. "Prediction of the Outcome of a Preoperative Chemotherapy in Breast Cancer Using DNA Probes that Provide Information on Both Complete and Incomplete Response". *BMC Bioinformatics* 9(149):1–17.

[54] Pedrosa, A. C. and B. W. Schmeiser. 1993. "Asymptotic and Finite-Sample Correlations between OBM Estimators". *Proceedings of the 1993 Winter Simulation Conference* 481–488.

[55] Peńa, D., and F. J. Prieto. 2001. "Multivariate Outlier Detection and Robust Covariance Matrix Estimation". *Technometrics* 43(3):286–310.

[56] Peng, J., P. Wang, N. Zhou, and J. Zhu. 2009. "Partial Correlation Estimation by Joint Sparse Regression Models". *Journal of the American Statistical Association* 104(486):735–746.

[57] Perron, F. 1992. "Minimax Estimators of a Covariance Matrix". *Journal of Multivariate Analysis* 43(1):16–28.

[58] Pourahmadi, M. 2000. "Maximum Likelihood Estimation of Generalised Linear Models for Multivariate Normal Covariance Matrix". *Biometrika* 87(2):425—435.

[59] Robbins, H. 1955. "A Remark on Stirling's Formula". *American Mathematical Monthly* 62:26–29.

[60] Rothman, A., P. Bickel, E. Levina and J. Zhu. 2008. "Sparse Permutation Invariant Covariance Estimation". *Electronic Journal of Statistics* 2:494–515.

[61] Rudemo, M. 1982. "Empirical Choice of Histograms and Kernel Density Estimators". *Scandinavian Journal of Statistics* 9(2):65–78.

[62] Satterthwaite, F. E. 1941. "Synthesis of Variance". *Psychometrika* 6(5):309–316.

[63] Schruben, L. W. 1983. "Confidence Interval Estimation Using Standardized Time Series". *Operations Research* 31:1090–1108.

[64] Scott, D. W. 2001. "Parametric Statistical Modeling by Minimum Integrated Square Error". *Technometrics* 43(3):274–285.

[65] Scott, D. W. 2004. "Partial Mixture Estimation and Outlier Detection in Data and Regression". *In Proceedings of Computational Statistics*, 297–306.

[66] Sharpe, W. F. (1994). "The Sharpe Ratio". *The Journal of Portfolio Management* 21(1):49–58.

[67] Shapiro, S. S., and M. B. Wilk. 1965. "An Analysis of Variance Test for Normality (Complete Samples)". *Biometrika* 52:591–611.

[68] Song, W. M., and B. W. Schmeiser. 1993. "Variance of the Sample Mean: Properties and Graphs of Quadratic-Form Estimators". *Operation Research* 41:501–517.

[69] Song, W. M., and B. W. Schmeiser. 1995. "Optimal Mean-Squared-Error Batch Sizes". *Management Science* 41:110–123.

[70] Steiger, N. M., E. K. Lada, J. R. Wilson, J. A. Joines, C. Alexopoulos, and D. Goldsman. 2005. "ASAP3: A Batch Means Procedure for Steady-State Simulation Analysis". *ACM Transactions on Modeling and Computer Simulation* 15(1):39–73.

[71] Steiger, N. M., and J. R. Wilson. 2001. "Convergence Properties of the Batch-Means Method for Simulation Output Analysis". *INFORMS Journal on Computing* 13(4):277–293.

[72] Stuart, A. M., and J. K. Ord. 1994. "Kendall's Advanced Theory of Statistics, Volume 1: Distribution Theory." 6th ed. London: Edward Arnold, Wiley.

[73] Sun, H., and H. Li. 2012. "Robust Gaussian Graphical Modeling via l_1 Penalization". *Biometrics* 68(4):1197–1206.

[74] Tafazzoli, A., N. M. Steiger, and J. R. Wilson. 2011. "N-Skart: A Nonsequential Skewness- and Autoregression-Adjusted Batch-Means Procedure for Simulation Analysis". *IEEE Transactions on Automatic Control* 56(2):254–264.

[75] Tafazzoli, A., and J. R. Wilson. 2011. "Skart: A Skewness- and Autoregression-Adjusted Batch Means Procedure for Simulation Analysis". *IIE Transactions* 43(2):110–128.

[76] Tafazzoli, A., J. R. Wilson, E. K. Lada, and N. M. Steiger. 2011. "Performance of Skart: A Skewness- and Autoregression-Adjusted Batch-Means Procedure for Simulation Analysis". *INFORMS Journal on Computing* 23:297–314.

[77] Tibshirani, R. (1996). " Regression Shrinkage and Selection via the Lasso". *Journal of the Royal Statistical Society, Series B* 58(1):267–288.

[78] von Neumann, J. 1941. "Distribution of the Ratio of the Mean Square Successive Difference to the Variance". *Annals of Mathematical Statistics* 12(4):367–395.

[79] Welch, P. D. 1987. "On the Relationship Between Batch Means, Overlapping Batch Means, and Spectral Estimation". *Proceedings of the 1987 Winter Simulation Conference*, 320–323.

[80] Whittaker, J. 1990. "Graphical Models in Applied Multivariate Statistics". New York: John Wiley & Sons.

[81] Wong, F., C. Carter, and R. Kohn. 2003. "Efficient Estimation of Covariance Selection Models". *Biometrika* 90(4):809–830.

[82] Wu, W. B., and M. Pourahmadi. 2003. "Nonparametric Estimation of Large Covariance Matrices of Longitudinal Data". *Biometrika* 90(4):831–844.

[83] Yuan, M., and Y. Lin. 2007. "Model Selection and Estimation in the Gaussian Graphical Model". *Biometrika* 94(1):19–35.

[84] Zou, H., and T. Hastie. 2005. "Regularization and Variable Selection via the Elastic Net". *Journal of Royal Statistical Society, Series B* 67(2):301–320.